REFERENCE MANUAL

Modular Composition System

FILE MANAGEMENT and TYPESETTING

Part No. 214979-003

SEPTEMBER 1982

FOREWORD

The attached manual describes the configurations and operating procedures of the **Modular Composition System**. Due to the many forms of applications, the recommended procedures should be followed.

This document is an instruction aid in the operation of the equipment and is subject to change. Compugraphic Corporation shall not be responsible for any claims or damages resulting from errors or omissions. Please report any errors or omissions to:

COMPUGRAPHIC CORPORATION

COPYRIGHT © SEPTEMBER 1982

Prepared by

Technical Product Support

200 Ballardvale Street

Wilmington, MA 01887

Type Style

Garth Graphic
Roman, Bold and Italic

Printed in U.S.A.

Table of Contents

Introduction 1-11
Typewriter Keyboard 6-11
  Precedent Characters 7
  Keyboard Arrangements 8-11
Programming the Terminal 12
Loading Fonts 12
Video Terminal Operation 12-23
  Status Lines 13-14
  Word Wrap 15
  Clear Display 15
  Time-Out 15
Editing Keypad 16-36
  Abort 16
  Cursor Positioning Keys 16-23
  Scrolling 21-23
Data Entry 25-27
  Cancel Character 24
  Cancel Word 24
  Composition Mode 25-26
  Edit Mode 27
  Local Compose Mode 27
Editing Functions 28-36
  Insert Mode 28
  Delete Character 28
  Delete Line 28
  Delete Page 29
  Define Block 30
  Delete Block 31
  Append Block 31
  Search and Replace 32-36
File Management 37-52
  Primary/Secondary Modes 38
  Dual Disk Operation 39
  Execute 40
  New File 40
  Display File Index 41
  Append Page 42-43
  Close File 44
  Protected Files 45
File Management (cont'd)
  Call Page 45-46
  Replace Page 47
  Insert Page 48-49
  Memory Full 48
  Insert Block 50
  Merge 51-52
Preparing Copy for Typesetting 53-99
  Parameter Display 54
  Line Length 55
  Font 56
  Point Size 57
  Set Size 58
  Line Space 59
  Line End Modes 60-61
  Hyphenation Program 62
  Alternate Hyphenation Programs 62
  Discretionary Hyphenation 62
  Justification 62
  Quadding Commands 63-64
  Spacebands (Wordspaces) 65
  Letterspace 66
  Fixed Space 67
Typesetter Commands 68-99
  Typesetter Command ON/OFF 68
  Precedence Keys 68
  Character Compensation 69
  AVCC 69
  Advance/Reverse Leading 70
  Plus/Minus Line Space 70
  Plus/Minus 1-Point Lead 70
  Insert Baseline Rule 70
Tab Programs 71-74
  Floating Tabs 71
  Call Tab 72
  Tab Return 72
  Deactivating Tabs 72
  Automatic Tabs 73-74
Insert Space 75
Insert Blank Space 75
Insert Character 76
UDKs 76
Indents 77-78
Cut Run-Arounds 79
No Flash (Non-Photo) 80
Flash Only 81
Manual Letterspacing 82-83
Pseudo Italic 84
Reverse Video 85
User Define Keys 86-94
Change Display Mapping 95
Typesetter Queue 96-97
Automatic Kerning 98
AVCC 99

APPENDIX A
  Pica/Cicero Conversion 101

APPENDIX B
  Copy Preparation 103
  Point/Picas 103
  Line Space 104-105
  Markup 106
  Copy Fitting 107-109
  Line Space Calculation 110

Quick Reference To:
  File Management 111-112
  System Commands 113
  Typesetter Commands 114
Vertical Justification Option 115-116

Index 117

Introduction

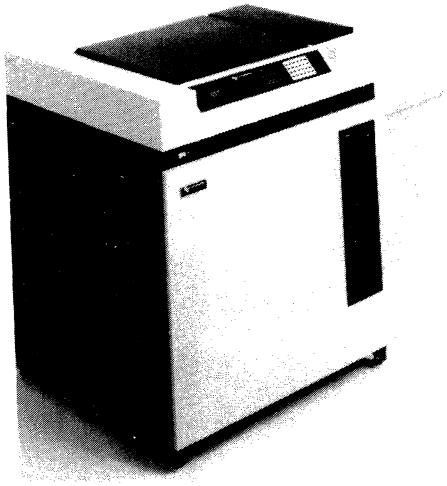
The **MCS FILE MANAGEMENT AND TYPESETTER REFERENCE MANUAL** describes the operating procedures of the MCS Video Terminal in conjunction with the resident phototypesetter. All facets of applications . . . file management, input and output are described. Although written in detail, the descriptions are for the purpose of general reference. For in-depth applications procedures the user should refer to the MCS Operators Training Manual (Part No. 214830).

Prior to operation, the operator must program the **MCS** *for the various applications used in the resident system.* The programming is used to create a **User Program Disk** which is used for everyday operation. The programming includes formatting of User Program Disks, implementation of the System Generator for the selection of options and the Font Utility Program. (See MCS Utility and Set Up Reference Manual, Part No. 214814).

VIDEO TERMINAL

The video terminal is used for data entry, file management and typesetting function commands. The terminal consists of a typewriter keyboard, two auxiliary keypads for editing and file management functions and a twelve-inch videoscreen for monitoring data. The screen displays 15 lines of data, two of which are status lines with the remaining 13 lines displaying text. Each line of text consists of 64 characters maximum having 256 variable alphanumeric characters or function symbols for display.

The status lines, the means by which the operator communicates with the system, verifies the status or errors associated with function commands. An indicator (cursor), is used to select the copy for editing and file management. As an aid to operation, typesetting function commands are displayed in half-intensity to distinguish them from the text. File data stored on the disk is output directly to the phototypesetter.



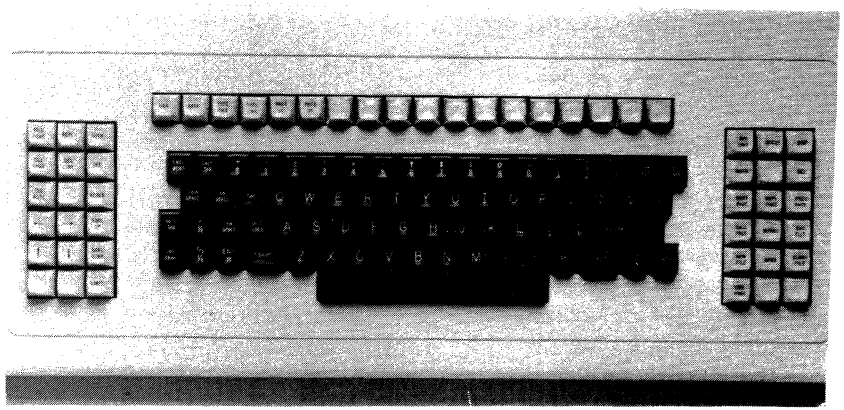
MCS8400 PHOTOTYPESETTER

TERMINAL KEYBOARD

The terminal keyboard has 123 key positions for the insertion of characters and functions. In addition, Control Keypads on the left and right side of the keyboard provide for editing and file management functions. During data entry, characters are displayed on the videoscreen. Where required, the character display can be changed to be compatible with various font complements. The keyboard is cable-connected to the controller and can be positioned for individual operator preference.

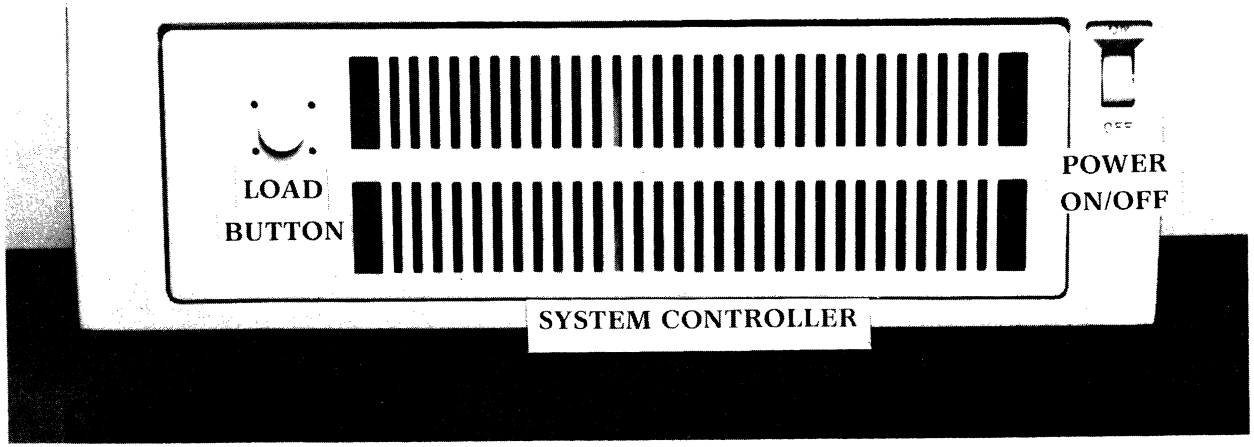
MINI FLOPPY DISK

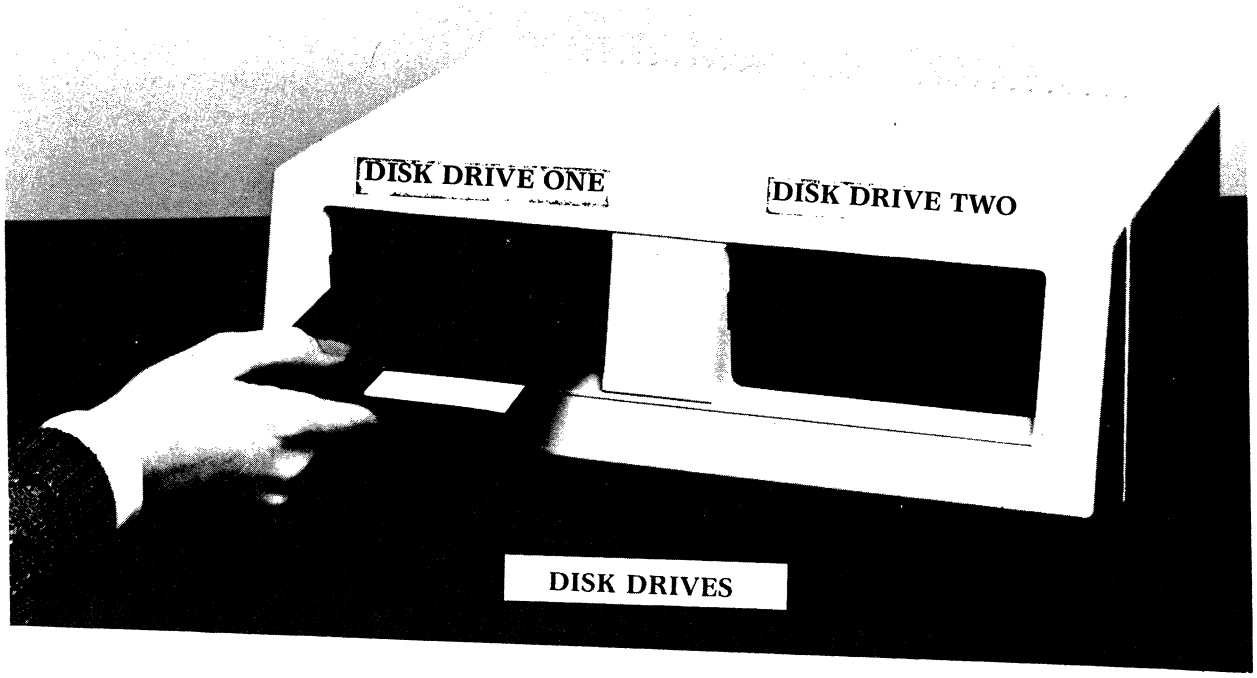
Data is stored on mini floppy disks. The number of files that can be stored on a disk is limited only by the capacity of the disk, approximately 160,000 characters. Files are divided into pages, with a maximum of 6000 characters per page. Only 832 characters can be displayed at any one time. However, scrolling functions, the vertical movement of text on the screen, permits viewing all characters stored in the terminal memory. *It should be noted that the mini disks are limited to MCS operation only.*



MCS TYPEWRITER KEYBOARD

System Controls





EDITING KEYPAD

The following is a brief description of the editing keys, located to the left of the alphanumeric keyboard:

ABORT—Cancels various system commands and file functions. Also removes the various displays from the VDT screen and returns the contents of the VDT memory to the screen.

LIMIT—Used in conjunction with the cursor movement and scrolling keys to 'jump' to the limit of the direction of the key.

↑—Moves the cursor in an upward direction. This is an auto repeat key and will continue operation while the key is depressed.

↓—Moves the cursor in a downward direction. This is an auto repeat key and will continue operation while the key is depressed.

SCRL DOWN—Scrolls copy down toward the start of the page. This is an auto repeat key and will continue operation while the key is depressed.

←—Moves the cursor in a left direction. This is an auto repeat key and will continue operation while the key is depressed.

→—Moves the cursor in a right direction. This is an auto repeat key and will continue operation while the key is depressed.

SCRL UP—Scrolls copy up toward the end of the page. This is an auto repeat key and will continue operation while the key is depressed.

INS MODE—Activates the insert mode of data entry, indicated by an I in front of the character count in message line two. This is an ON/OFF toggle keyswitch. The next activation of the key will return the VDT to the overstrike mode of data entry.

BLOCK—Used in conjuction with the Define, Append and Delete keys to define, append or delete blocks of copy.

DEL CHAR—Deletes the character located at the cursor position. This is an auto repeat key and will continue operation while the key is depressed.

DEL LINE—Deletes a line of copy from the VDT memory. This is a two-keystroke function. The first activation of the key underscores all characters from the cursor to the end of the screen line; the second activation deletes all the underscored characters.

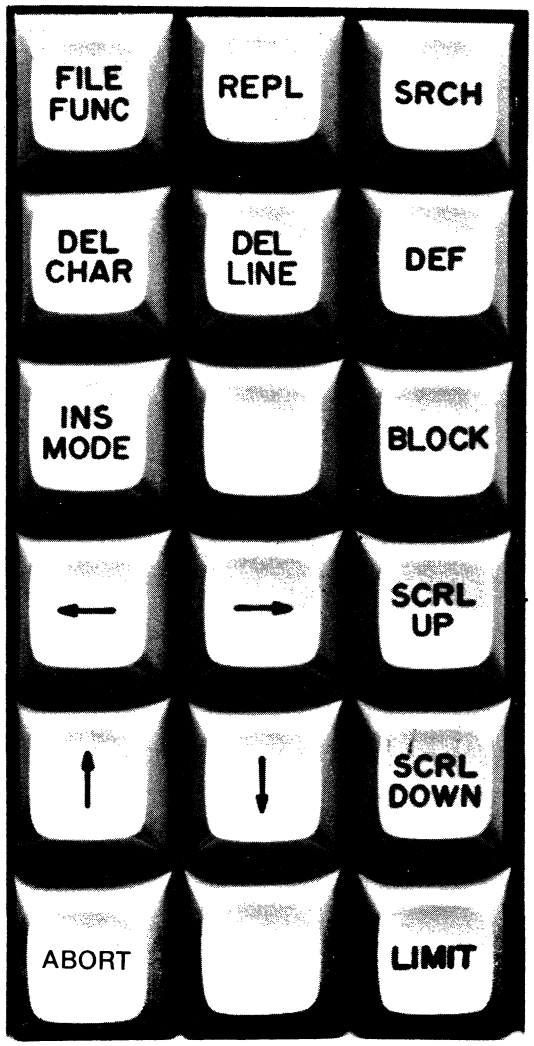
DEF—Used in conjunction with the Block and User Define Keys to define blocks of copy or the User Define Keys.

FILE FUNC—Used with Compose/Execute to display all active typesetting parameters. Used with the Search and Replace keys to search an entire file. Also used with various MCS options.

REPL—Used with the Search key for a search and replace operation. Up to 16 characters may be entered in the replace string.

SRCH—Used to search the VDT memory for an operator definable character string of up to 16 characters.

EDITING CONTROL KEYPAD



FILE MANAGEMENT KEYPAD

The following is a brief description of the file management keys, located to the right of the alphanumeric keyboard.

SYSTEM COMMAND—Used with abbreviated function descriptions to load fonts (LFNT), display maps (DMAP), delete fonts (DFNT), etc.

QUEUE—Used with abbreviated function descriptions and file names to transfer files to the typesetter (TY), printer (PR) or preview (PV) for processing.

DISPLAY—Used with abbreviated function descriptions to display the disk index (DI), resident fonts (FT) and various queues (TY, PR, PV). The Abort key is used to remove the display function and restore the previous screen display.

INSERT—Used to transfer data from the screen memory (up to the cursor) to the addressed file and page. Subsequent pages numbers are incremented by one.

DELETE—Used to delete the addressed page of the open file from the disk. Also used to delete data from the screen memory.

NEXT PAGE—Used to access the page which follows the current page in the addressed file. Should be performed in the compose mode to preserve parameters.

REPLACE PAGE—Used to transfer all data in the screen memory to the disk, which overprints the original version; then calls the next page to the screen.

PREVIOUS PAGE—Used to call the preceding page to the screen.

CALL PAGE—Used to open an existing file and call the first page to the screen. Can be used to address any page in the file by page number but continuity of parameters is not maintained and becomes an operator responsibility.

MERGE—Used to transfer a page from the same file or another file to the screen and insert it at the cursor position.

SECONDARY FILE—Used to communicate with another file to transfer a page from that file to the screen. Can also be used to transfer a page or a block from the current file to another file.

NEW FILE—Used to establish a file name on a disk for the storage of new data.

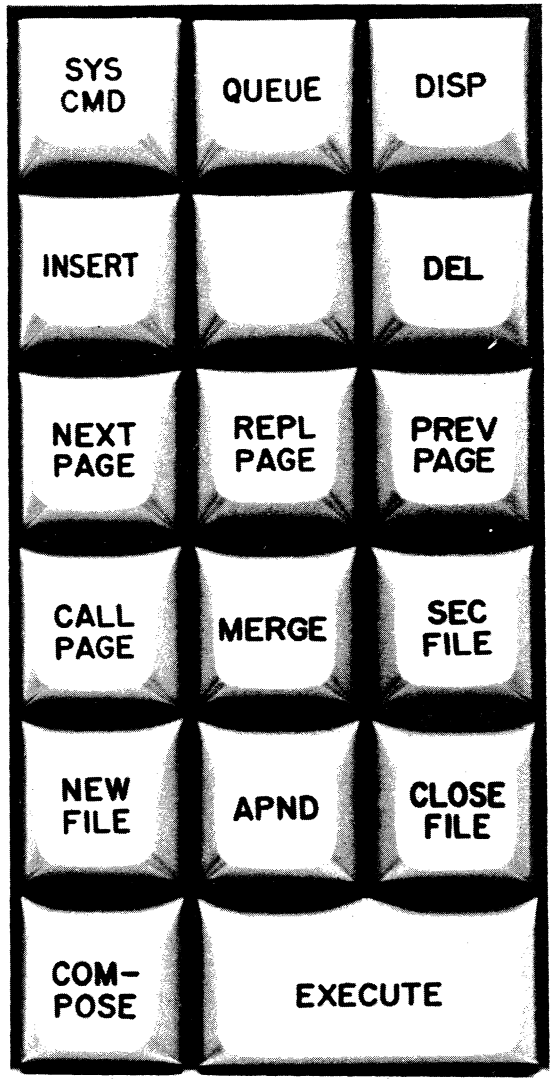
APPEND—Adds data preceding the cursor to the end of the addressed file. Most frequently used when the keyboard memory is nearly full to transfer data preceding the cursor to the end of the current file.

CLOSE FILE—Must be used after entry of new data or editing of an existing file has been completed and before another file can be opened.

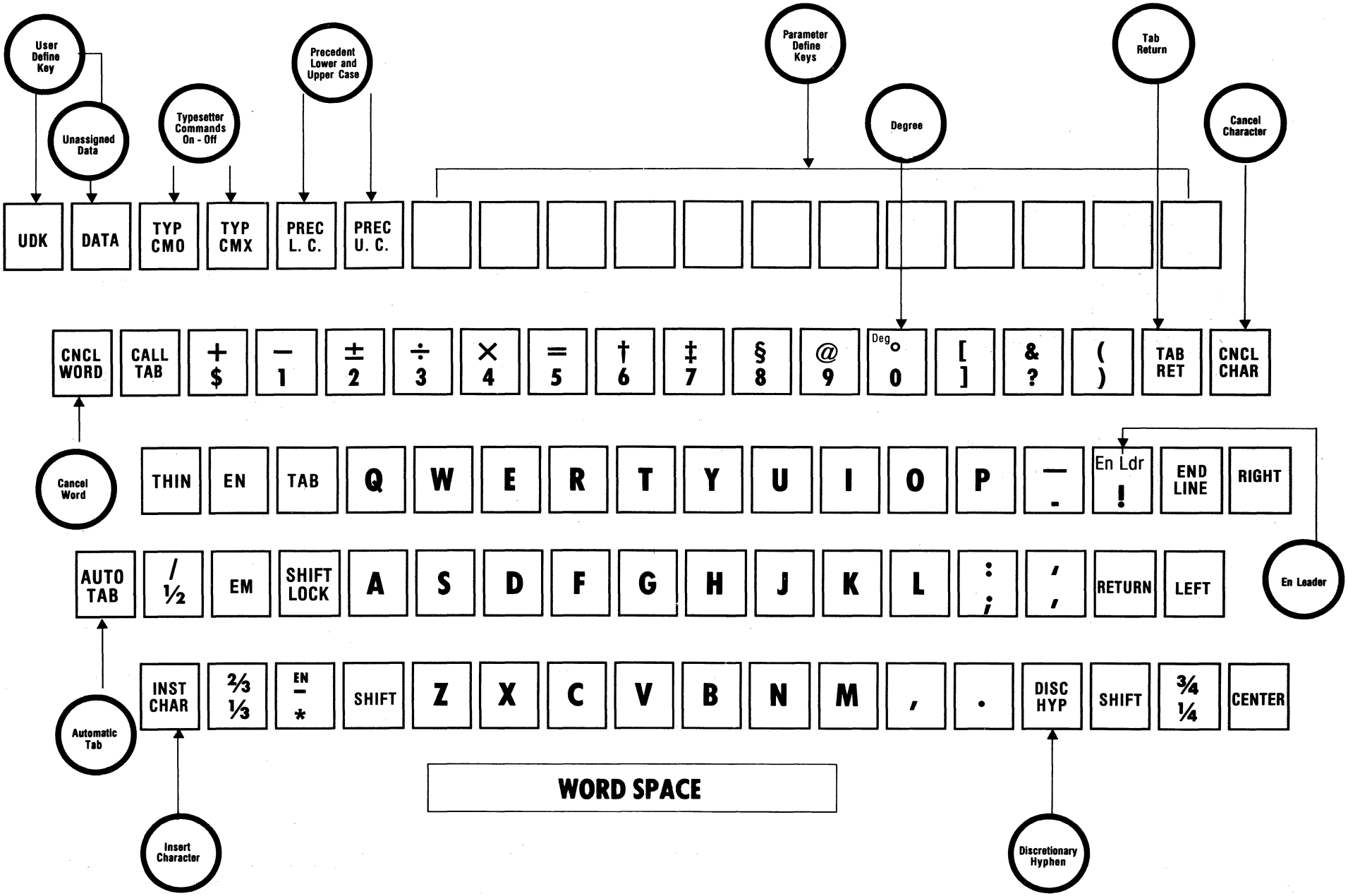
COMPOSE—Causes the system to scan all data from the start of the page up to the character preceding the cursor. Parameters are updated. Character widths are counted and line-endings are displayed. Also used after File Function and before Execute to display parameter index.

EXECUTE—Used to signify completion of file management and other functions.

FILE MANAGEMENT CONTROL KEYPAD



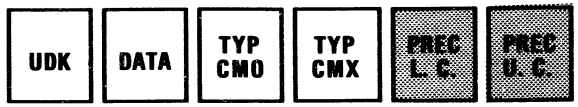
Typewriter Keyboard



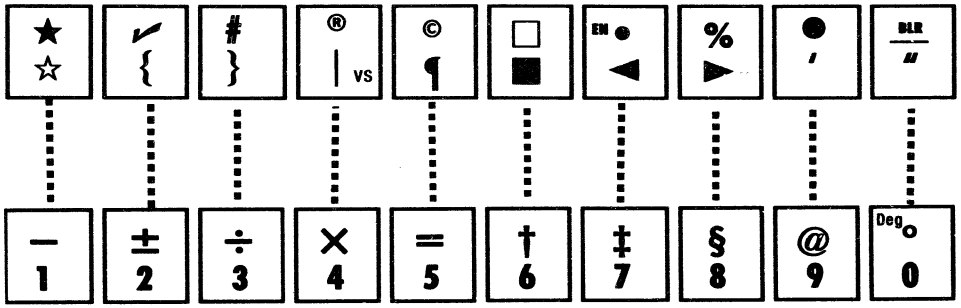
Typewriter Keyboard

PRECEDENCE CHARACTERS

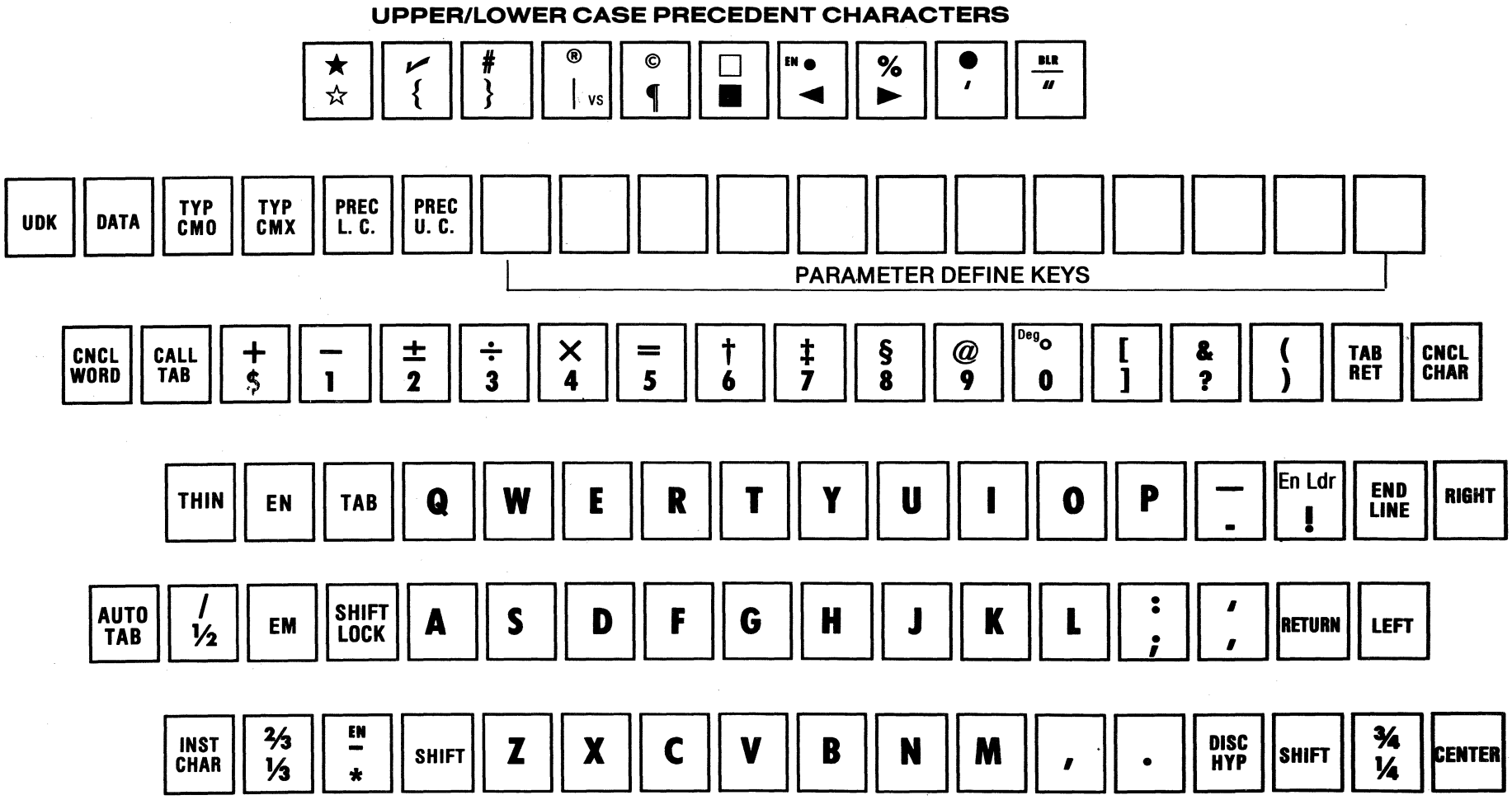
The **MCS** has a standard typewriter keyboard which directly accesses 98 characters. However, each font complement consist of 128 characters. To access the 20 additional characters, two function keys located on the top row of the keyboard are utilized. Designated as *Precedence Upper Case* and *Precedence Lower Case,* they are used in conjunction with the figures 1 to 0 on the keyboard. Pressing the Precedence key, followed by a figure key, will access the character associated with the font complement. For example, the Precedence Lower Case key followed by the figure 1 calls for the Open Star (☆) in the 501 complement. Each Precedence key produces a message in the status line identifying which key was used, *PrecedenceUC* or *PrecedenceLC.* When a figure key is struck, the selected character image appears in the text stream and the message is cleared from the status line.



UPPER/LOWER CASE PRECEDENCE CHARACTERS

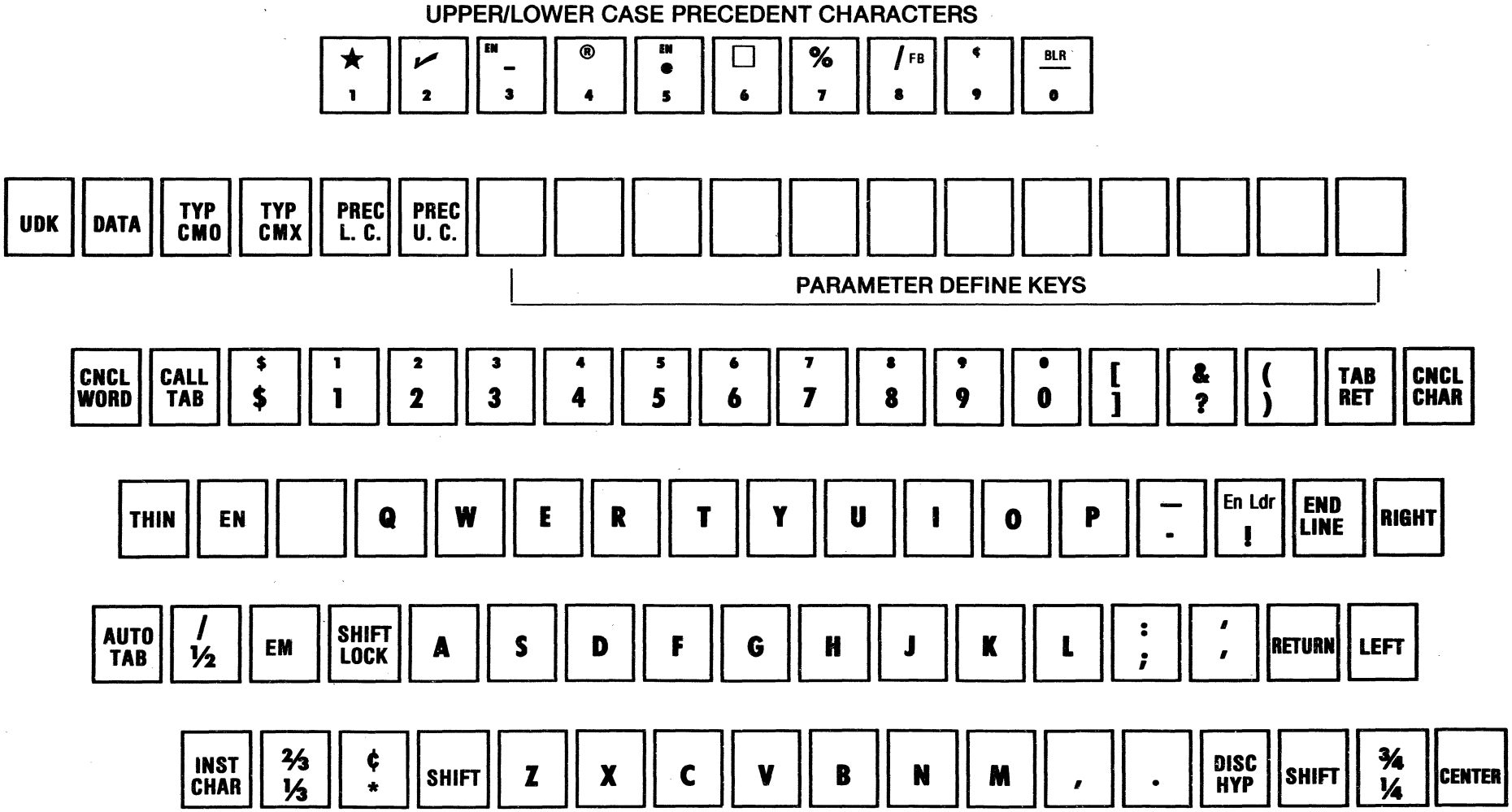


Keyboard Arrangement 501



ABBREVIATIONS:
EN•—En Bullet
EN––En Dash
VS—Vertical Separator
BLR—Baseline Rule
En Ldr—En Leader
Deg—Degree

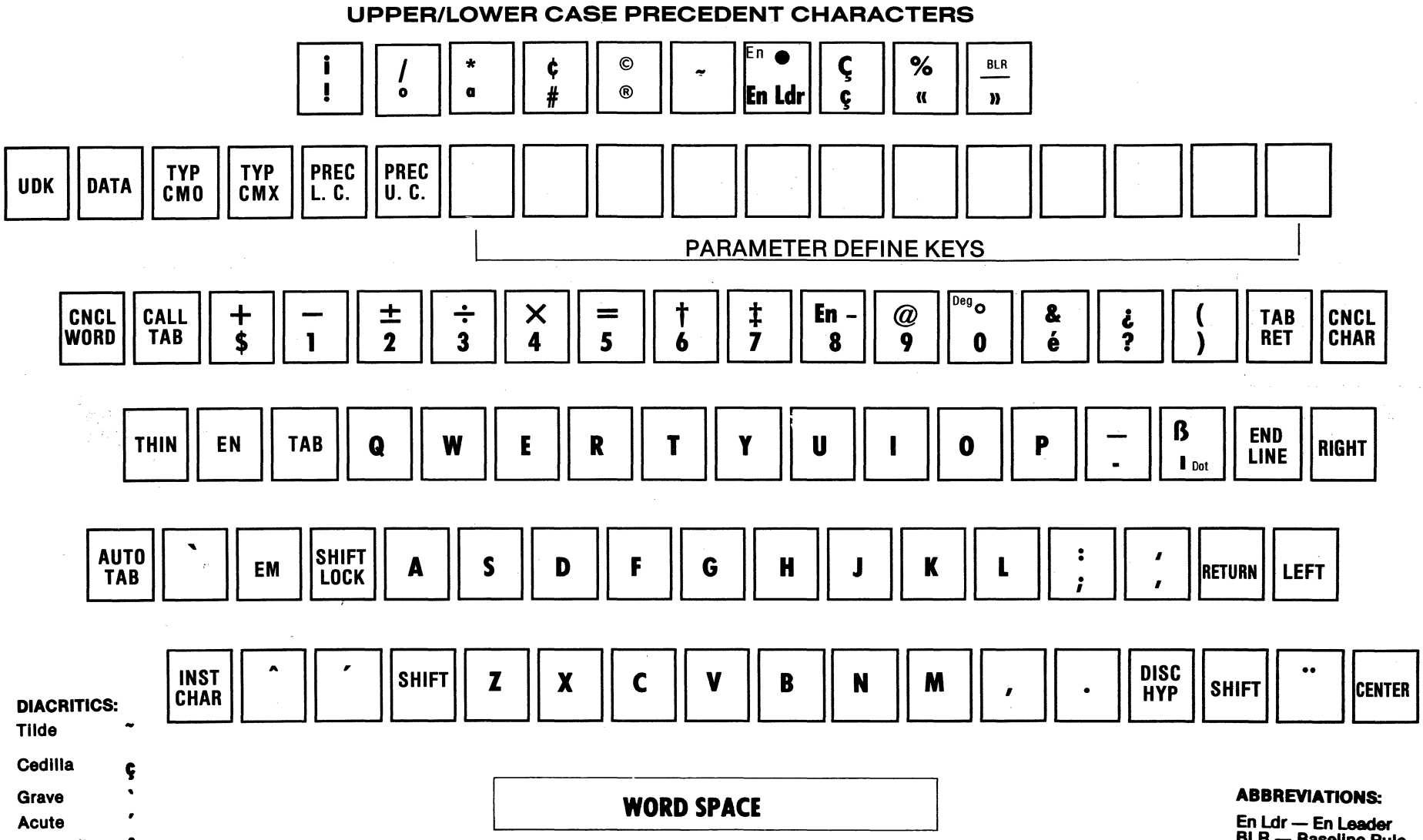
Keyboard Arrangement 500



ABBREVIATIONS:

En - — En Dash
FB — Fraction Bar
BLR — Baseline Rule
En Ldr — En Leader

Keyboard Arrangement 549



DIACRITICS:

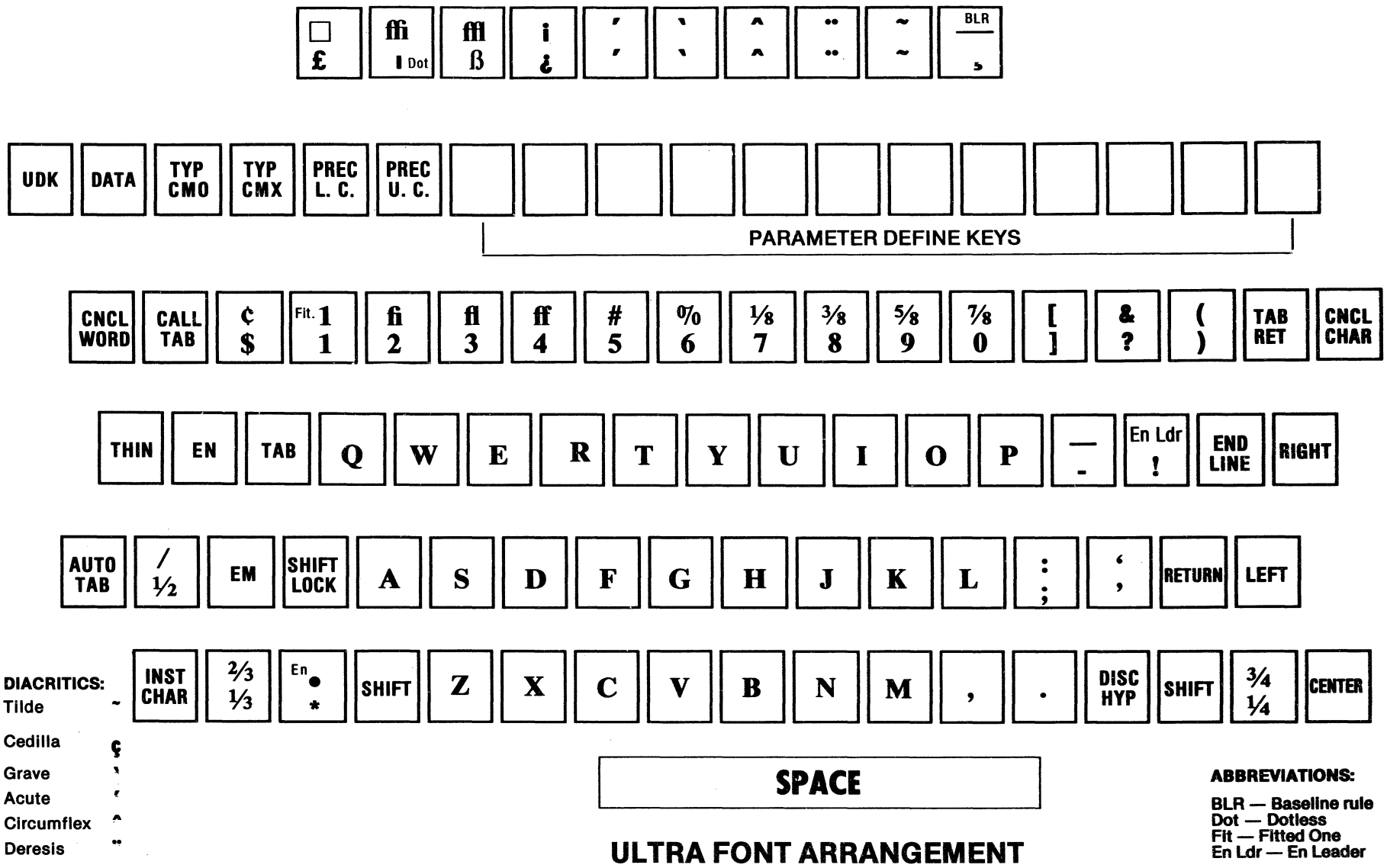
Tilde	~
Cedilla	ç
Grave	`
Acute	´
Circumflex	^
Deresis	••

ABBREVIATIONS:

En Ldr — En Leader
BLR — Baseline Rule
En - — En Dash
Deg — Degree

Keyboard Arrangement 1614

UPPER/LOWER CASE PRECEDENT CHARACTERS



DIACRITICS:

Tilde ~
Cedilla ¸
Grave `
Acute ´
Circumflex ^
Deresis ¨

ULTRA FONT ARRANGEMENT

ABBREVIATIONS:

BLR — Baseline rule
Dot — Dotless
Fit — Fitted One
En Ldr — En Leader

PROGRAMMING THE MCS TERMINAL

Before an MCS terminal can be used to process text material, it must be programmed with the appropriate User Program Disk(s). This loads the various functional programs and configures the terminal for communication with the appropriate input/output devices. The program remains active until power is turned off. If a terminal is powered down at the end of each shift, it must be re-programmed at the start of the next shift.

The following sequence should be followed:

1) Turn the Controller on. No disks should be engaged in terminal disk drives.
2) Insert UPD program disk Vol. I in a terminal disk drive. On multi-drive systems, any drive can be used.
3) Press the Reset button on the Controller.
4) After a brief pause, the message "Vol. I is Loading" appears on the screen of the terminal addressed to Drive 1.
*5) When Vol. I has been loaded, the message will be replaced with "Insert Vol. II and hit Return".
6) Insert Vol. II; press Return on the Drive 1 terminal.
7) When the program has been loaded, the message "No Fonts" appears in the message line.
8) Load fonts either individually by number or by font dress name (see Utility and Set Up Procedures Manual).
9) Insert a work disk in a terminal disk drive. The terminal is now available to process data.

NOTE: Systems which utilize only one or two options require only a single program disk (Volume I); steps 5 and 6 do not apply.

FONT-LOADING PRECAUTIONS

The operator should be aware that when loading fonts into the terminal memory, the system loads character description tables for only the FIRST TWO* font complements encountered. These tables are used in conjunction with the hyphenation program to determine hyphenation points for a particular character complement, such as 500, 501 or 549. Any additional layouts encountered during the loading process will default to the first layout entered and the message line will display *"Default Complement"* as fonts are loading.

If the basic layouts are the standard 500/501 complements, then the first fonts loaded must be 500 and 501. Any other layouts that follow will not be included in the hyphenation descriptor table, as indicated by *Default Complement.*

This also applies when creating font dress files. For example, if the base hyphenation program is French, one of the first fonts loaded should be a 549 or 1614 complement. This will ensure proper hyphenation breaks when that layout is being used for composition.

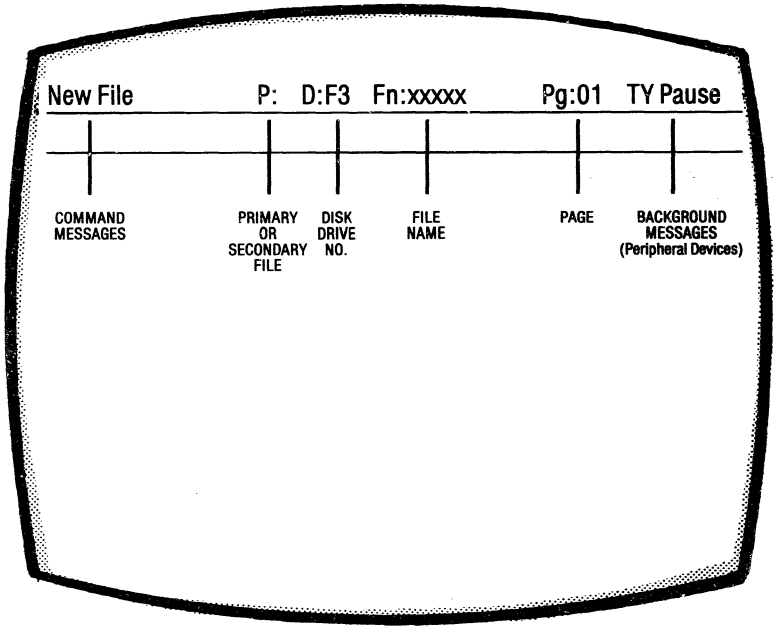
Fonts stored in the terminal can be erased and reloaded in a different sequence where necessary. An individual font can be erased by pressing the SYS CMD key, typing D/F/N/T and the font ID number, followed by EXECUTE. If the entire font storage is to be erased, the Edit Mode must be enabled by the FILE FUNC key, typing the letters E/D/I/T, followed by the EXECUTE key. Fonts can then be erased by the SYS CMD key, typing the letters C/F/M/E/M, and the EXECUTE key.

*If the Extra Font Option has been exercised (32 fonts rather than 16), tables will be loaded for FOUR font complements and the default message will not appear until a fifth complement is loaded.

STATUS LINES

The video terminal is the operator's medium for communication with the system. As text or functions are entered from the keyboard, the videoscreen indicates the current operating status by displaying messages and parameters. In addition, 13 lines are provided for the display of text.

The status lines prompt for functions, alert the operator to errors and verify typesetting parameters associated with these commands. Status messages are displayed at full intensity and are separated from text by ruled lines. If a system error occurs, it will be indicated by an audible tone. The cursor will be positioned on the error and a description of the problem will be displayed in the message area.



Status Line One

[New File]	Current File Management function command messages.
[P] [S]	Primary or Secondary File selected. Secondary indicates selection of an interim file, or file data to be used interactively with the Primary File.
[D:F#] or **[D:H#]**	Disk Drive Number: F for Floppy Disk Drives; H for Hard Disk Drives.
[Fn:]	Indicates File Name of current Primary or Secondary File.
[Pg:]	Indicates Page Number of current Primary or Secondary File.
[TY Pause]	Input/Output messages relating to the status of background functions (peripheral devices).

Status Line Two

[$P_S^=$] Indicates currently selected type size in point and half-point increments.

[$L_S^=$] Indicates the currently selected line space (Leading) in points and quarter-points.

[$F_T^=$] Indicates currently selected font number.

[$L_L^=$] Indicates the currently selected line length (measure) in picas and points.

[A] [M] [S] Line-ending symbols. Indicate the method by which line-ending decisions are determined: automatic, manual, semi-automatic.

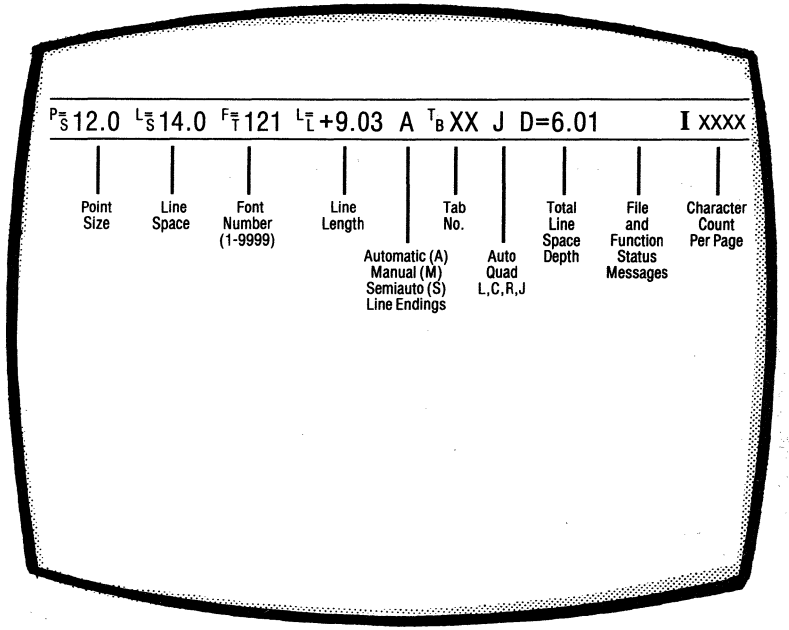
[$T_B^=$] Indicates the currently selected tab number.

[L, C, R, J] Automatic line-ending modes: quad left, quad center, quad right or justified.

[D = xxx.xx] Indicates the total depth in picas and points.

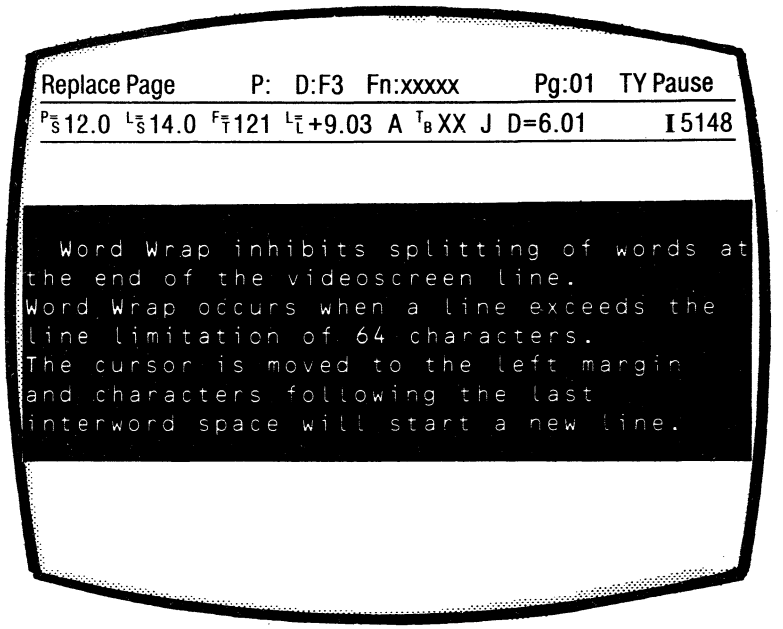
[Complete] File status message indicating completion of functions.

[I xxxx] I = Insert Mode indicator.
x's = Character count of current page.


P=S 12.0 L=S 14.0 F=T 121 L=L +9.03 A T=B XX J D=6.01 I xxxx
Point Size
Line Space
Font Number (1-9999)
Line Length
Automatic (A) Manual (M) Semiauto (S) Line Endings
Tab No.
Auto Quad L,C,R,J
Total Line Space Depth
File and Function Status Messages
Character Count Per Page

WORD WRAP

The purpose of Word Wrap is to prevent the division of words at the end of a line on the screen during data entry or recomposition. Word Wrap occurs when a line exceeds the 64-characters-per-line limitation. All characters following the last interword space or function delimiter will be carried over and start a new line. The cursor moves with the word or symbol that will start the new line. If a screen line does not contain a word space or a function, no true Word Wrap can occur. The 65th character will be carried over to start a new line. If a function occurs on such a line, Word Wrap will take place immediately at the end of the function.


Replace Page P: D:F3 Fn:xxxxx Pg:01 TY Pause
PS12.0 LS14.0 FT121 LL+9.03 A TB XX J D=6.01 I5148
Word Wrap inhibits splitting of words at the end of the videoscreen line.
Word Wrap occurs when a line exceeds the line limitation of 64 characters.
The cursor is moved to the left margin and characters following the last interword space will start a new line.

CLEAR DISPLAY

All data currently stored in the terminal memory can be erased at any time. All data on the screen and in memory will be deleted. *Data on the disk is not affected.* If addressed to an open file, typesetting parameters and the depth count in effect at the start of the page will be re-established after the display is cleared. The system will remain addressed to the original page.

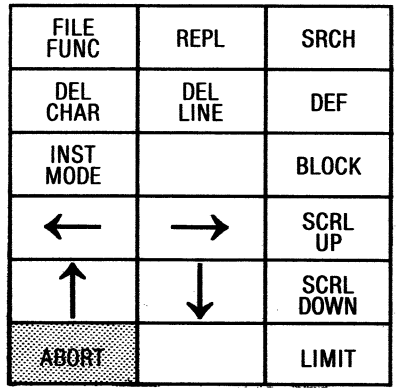
Procedure: *Delete/Display/Execute*

TIME-OUT

If there is no activity on the screen for a period of five minutes, an automatic Time-Out will occur and the screen will go blank. The purpose of the Time-Out is to prolong the life of the videotube. Data can be restored to the screen by depressing any key, preferably a cursor key.

EDITING KEYPAD

The Editing Keypad functions control the placement of text data on the videoscreen. Editing functions are used for revisions, i.e., deletion and insertion of job data. This is accomplished by scrolling and utilizing a cursor to locate and specify data.

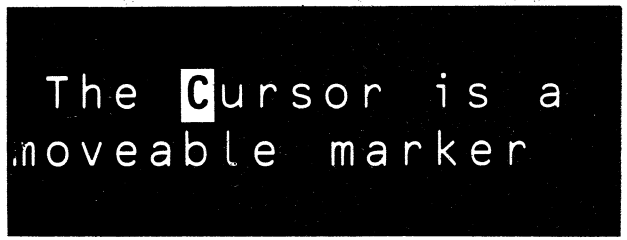


ABORT

The *Abort* function is used to cancel a function command. Striking *Abort* (Editing Keypad) prior to keying the command function *Execute* cancels the command.

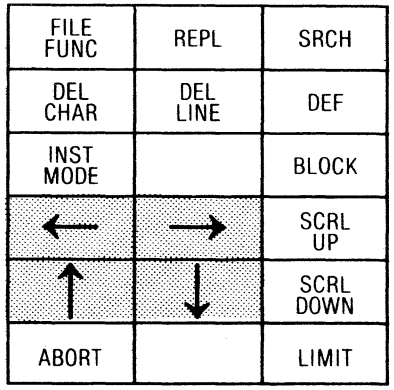
CURSOR

The cursor is a moveable marker that can be positioned to indicate a specific location or character on the videoscreen. The cursor, shown in reverse video, appears automatically when opening a file and is utilized for editing and/or management of file data. Data is entered from the keyboard and inserted at the position of the cursor. The cursor is then moved one position to the right to allow for each entry. If the data is a line ending function, the remainder of the screen is left blank and the cursor positioned at the beginning of the next line. At any other time, the horizontal or vertical movement of the cursor is controlled by cursor positioning keys located on the Editing Keypad. *Pressing any one of the keys moves the cursor in the direction of the arrow. When positioned on the last character it does not move.*



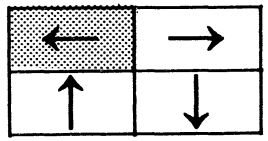
CURSOR POSITIONING KEYS

The cursor positioning keys are used for editing and file management. They control the movement and location of the cursor. The cursor can be positioned horizontally or vertically and moves one position for each depression of a cursor key. If the key is held down, movement of the cursor is repeated. While in the Compose Mode, character widths are recalculated with any movement of the cursor. Moving the cursor in a reverse direction beyond any two line-endings (including tabs) will put the system in the Edit Mode.



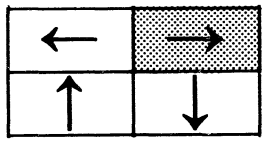
CURSOR LEFT

When Cursor Left (←) is pressed, the cursor moves to the left one position. If placed at the beginning of the line, the next depression will move the cursor to the end of the previous line. When the cursor is on the first character of the first line, the line will scroll down and the cursor will appear at the end of the newly displayed line. If the cursor is on the first character of the first line of a page, the cursor will not move.

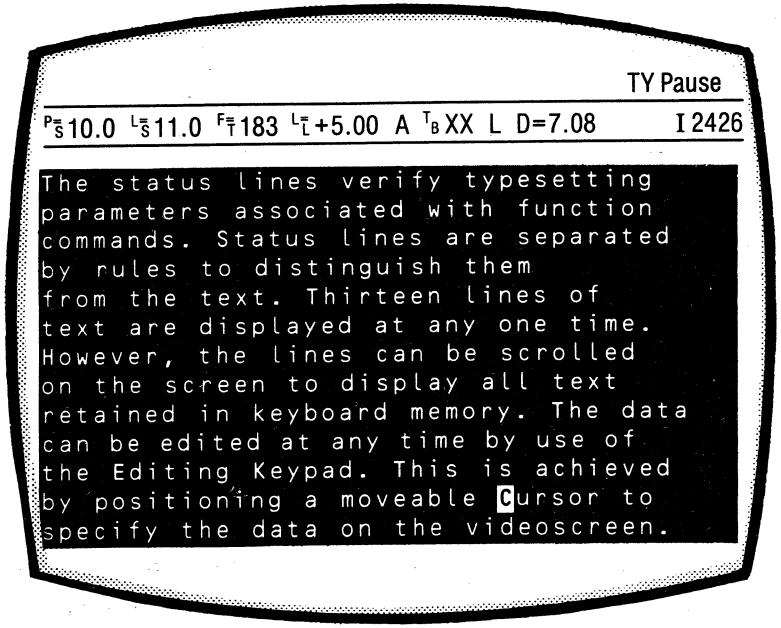


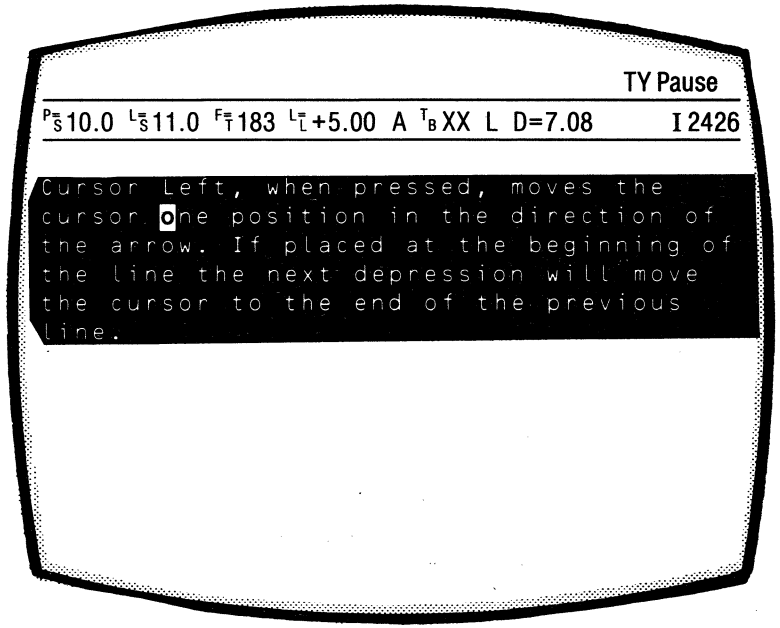
CURSOR RIGHT

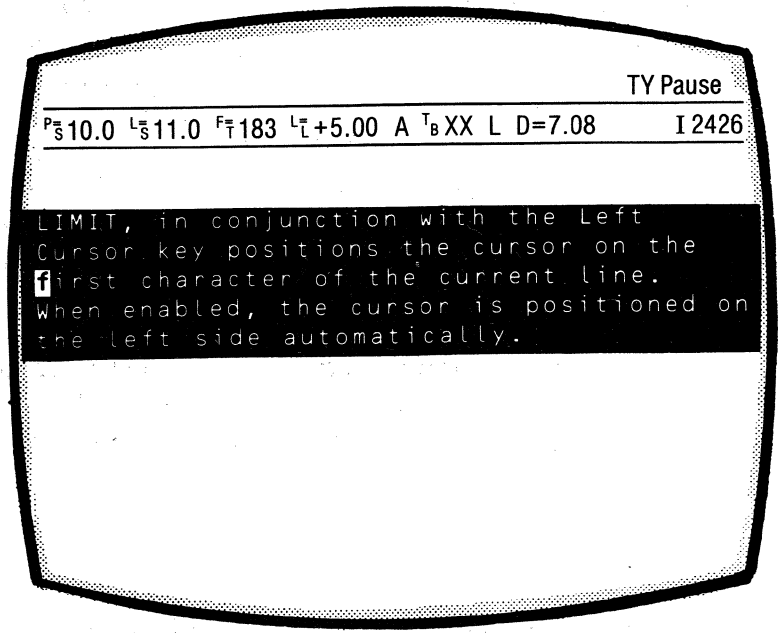
When Cursor Right (→) is pressed, the cursor moves to the right one position. If placed at the end of the line, the next depression will move the cursor to the beginning of the following line. When moved beyond the right-most character on the last line of the display, the screen will scroll up one line and the cursor will be positioned at the start of the next line. If placed after the last character of the last line of the page, the cursor will not move.

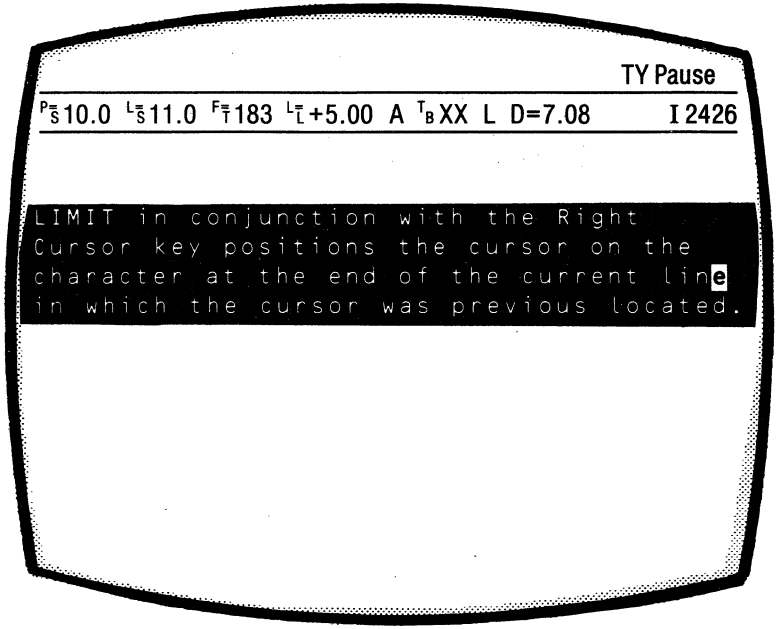


The right cursor is also used to identify the end of a field or to access the close Control symbol when supplying data for a programmed key.


TY Pause
P=S10.0 L=S11.0 F=T183 L=L+5.00 A TB XX L D=7.08 I 2426
The status lines verify typesetting parameters associated with function commands. Status lines are separated by rules to distinguish them from the text. Thirteen lines of text are displayed at any one time. However, the lines can be scrolled on the screen to display all text retained in keyboard memory. The data can be edited at any time by use of the Editing Keypad. This is achieved by positioning a moveable Cursor to specify the data on the videoscreen.

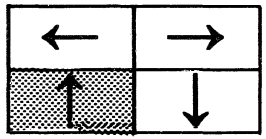

TY Pause
P=S10.0 L=S11.0 F=T183 L=L+5.00 A TB XX L D=7.08 I 2426
Cursor Left, when pressed, moves the cursor one position in the direction of the arrow. If placed at the beginning of the line the next depression will move the cursor to the end of the previous line.


TY Pause
P=S10.0 L=S11.0 F=T183 L=L+5.00 A TB XX L D=7.08 I 2426
LIMIT, in conjunction with the Left Cursor key positions the cursor on the first character of the current line. When enabled, the cursor is positioned on the left side automatically.


TY Pause
P=S10.0 L=S11.0 F=T183 L=L+5.00 A TB XX L D=7.08 I 2426
LIMIT in conjunction with the Right Cursor key positions the cursor on the character at the end of the current line in which the cursor was previous located.

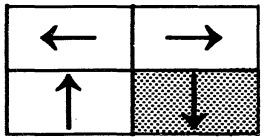
CURSOR UP (↑)

When the *Cursor Up* key is pressed, it moves the cursor one position in the direction of the arrow, placing it on the previous line. If positioned at the top of the screen, the next depression will scroll the top line down. If the cursor is on the first line of the page, it will not move.



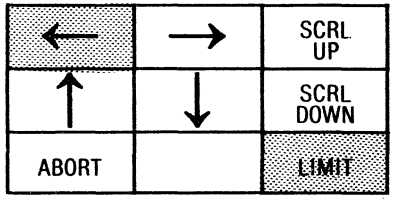
CURSOR DOWN (↓)

When the *Cursor Down* key is pressed, the cursor is moved one position in the direction of the arrow, placing it on the succeeding line. If positioned at the bottom of the screen, the next depression will scroll the screen up one line. When the cursor is on the last line of the page, it will not move.



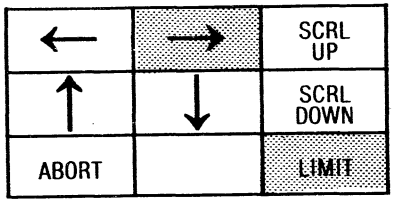
BEGINNING OF LINE

When the *Limit* function key is followed by the *Cursor Left* key, it immediately moves the cursor to the first character of the current screen line. This is a time-saving procedure when editing copy.



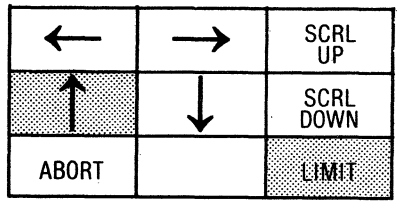
END OF LINE

When *Limit* is followed by *Cursor Right,* it immediately moves the cursor to the last character of the current screen line and is a recommended procedure during editing.



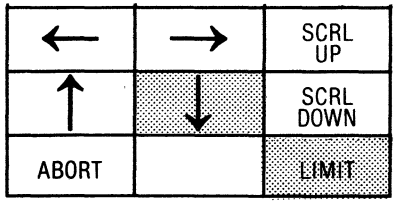
SKIP UP

The function LIMIT, followed by the Cursor Up (↑) key moves the cursor up to the beginning of the first line on the display. There is no other change in the status of data displayed on the screen. If undisplayed data occurs prior to the cursor position and if the key sequence is repeated, data will automatically scroll down ten lines. The cursor will remain in its previous position on the top line. Moving the cursor beyond two previous line endings activates the Edit Mode of operation.

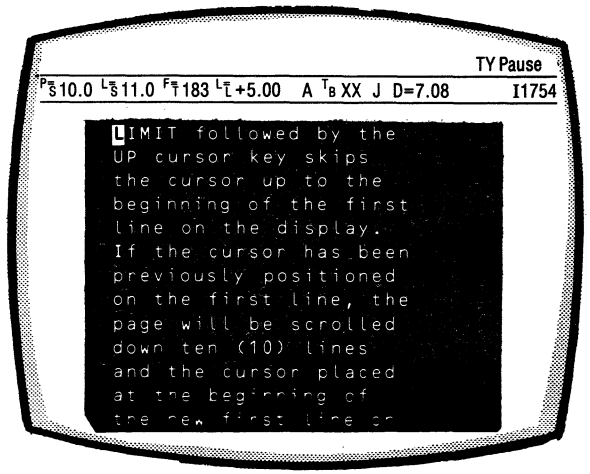


SKIP DOWN

The function LIMIT, followed by the Cursor Down (↓) key moves the cursor down to the beginning of the last line on the screen. If further data occurs after the cursor location and if the key sequence is repeated, data will automatically scroll up ten lines. The cursor will remain in its previous position on the bottom line.



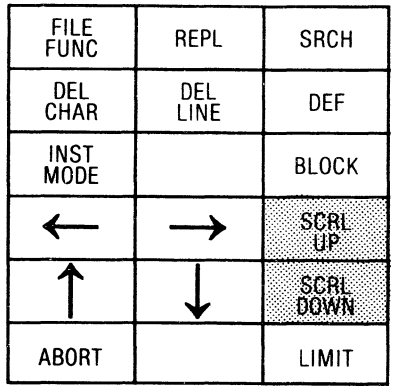
Both skip functions are designed to provide immediate access to the top and bottom of the videoscreen.

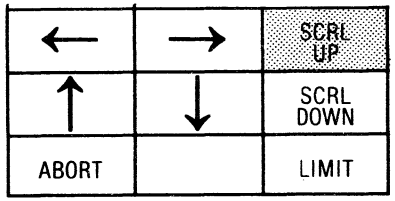


SCROLLING

Scrolling is the movement of data up and down on the videoscreen. A minimum of 13 screen lines must be occupied before scrolling can occur. Although the terminal can store up to 6,000 characters, only 13 lines (832 characters) can be displayed at any one time. Scrolling occurs automatically during data entry when text exceeds the screen line limitation. At that point, the top line rolls off the screen and the cursor moves to a blank line that is added at the bottom. This is repeated until the page is completed.

Scrolling is also achieved by the *Scroll Up* and *Scroll Down* keys on the Editing Keypad. A single depression of either key moves screen data up or down one line. If either key is held down, data remaining in memory will be displayed line by line. The cursor moves with its original line until that line reaches the top or bottom of the screen. The mode of operation, Compose or Edit, will be retained during scrolling except that the Edit Mode will be invoked by any scroll or cursor activity that causes the cursor to move back beyond two previous line endings.

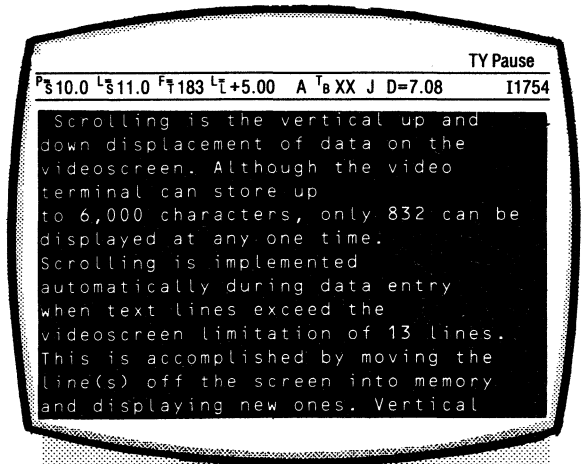


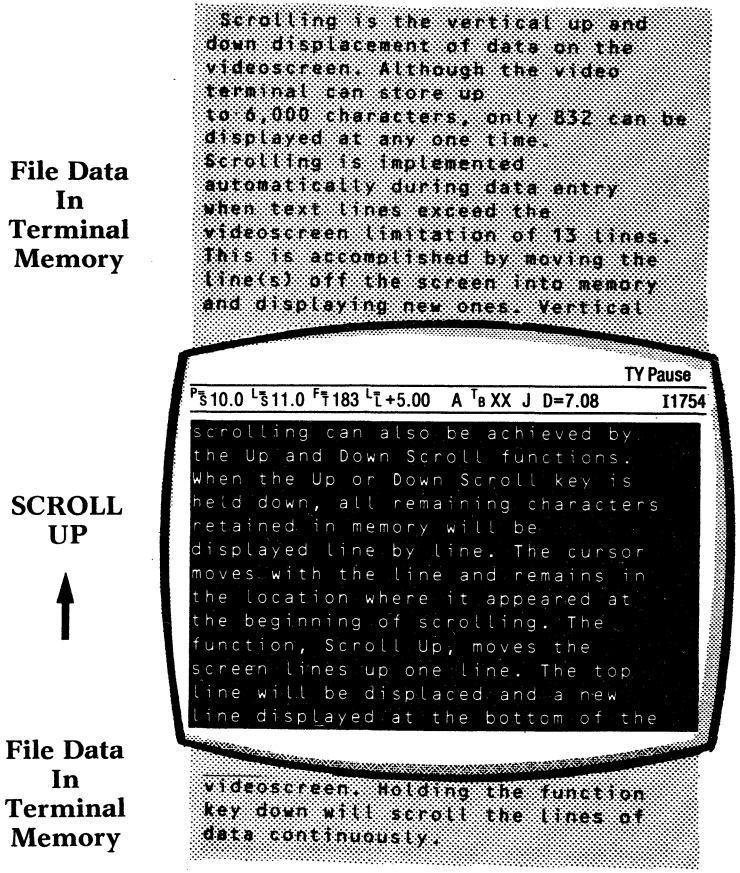


Scroll Up

Each depression of the Scroll Up key moves data on the screen up one line. Text in the top line will be displaced and a new line will be added at the bottom of the screen. If the Scroll Up key is held down, lines of text will be scrolled up continuously. Scrolling will stop when the last line of the page appears on the screen.

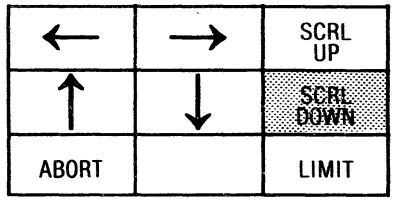
SCROLLING


SCROLL UP
Videoscreen (13 Lines)
File Data In Terminal Memory
TY Pause
PS10.0 LS11.0 FT183 LT+5.00 A TB XX J D=7.08 I1754
Scrolling is the vertical up and down displacement of data on the videoscreen. Although the video terminal can store up to 6,000 characters, only 832 can be displayed at any one time. Scrolling is implemented automatically during data entry when text lines exceed the videoscreen limitation of 13 lines. This is accomplished by moving the line(s) off the screen into memory and displaying new ones. Vertical
scrolling can also be achieved by the Up and Down Scroll functions. When the Up or Down Scroll key is held down, all remaining characters retained in memory will be displayed line by line. The cursor moves with the line and remains in the location where it appeared at the beginning of scrolling. The function, Scroll Up, moves the screen lines up one line. The top line will be displaced and a new line displayed at the bottom of the videoscreen. Holding the function key down will scroll the lines of data continuously.


File Data In Terminal Memory
SCROLL UP
File Data In Terminal Memory
Scrolling is the vertical up and down displacement of data on the videoscreen. Although the video terminal can store up to 6,000 characters, only 832 can be displayed at any one time. Scrolling is implemented automatically during data entry when text lines exceed the videoscreen limitation of 13 lines. This is accomplished by moving the line(s) off the screen into memory and displaying new ones. Vertical
TY Pause
PS10.0 LS11.0 FT183 LT+5.00 A TB XX J D=7.08 I1754
scrolling can also be achieved by the Up and Down Scroll functions. When the Up or Down Scroll key is held down, all remaining characters retained in memory will be displayed line by line. The cursor moves with the line and remains in the location where it appeared at the beginning of scrolling. The function, Scroll Up, moves the screen lines up one line. The top line will be displaced and a new line displayed at the bottom of the
videoscreen. Holding the function key down will scroll the lines of data continuously.

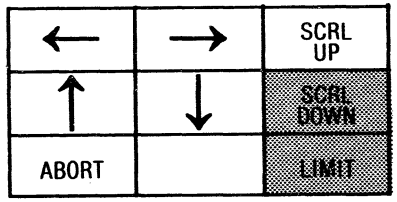
SCROLL DOWN

When more than 13 screen lines have been occupied, employing the *Scroll Down* function moves the screen lines down one position. The last line of text will be removed and a new line will be displayed at the top of the screen. If *Scroll Down* is held down, text will be scrolled down continuously. Scrolling will stop when the first line of the page has been displayed.



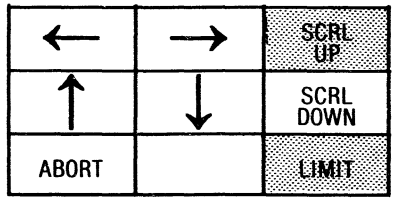
BEGINNING OF PAGE)

In order to rapidly view the first lines of a page, *Limit* and *Scroll Down* will immediately display the first 13 lines. The cursor will be positioned on the first character of the first line.



END OF PAGE

To rapidly view the last lines of a page, *Limit* and *Scroll Up* will immediately display the last 13 lines. The cursor will be positioned after the last character on the page.



CANCEL CHARACTER

All data is input via the typewriter keyboard. If a typing error occurs during entry, it can be quickly corrected with the Cancel Character key. When utilized, *the character to the left of the cursor will be cancelled.* If the cursor is on the first character of a line, the last character of the previous line will be deleted, including a return function. If the cursor has been positioned on the first character of the first display line, data will be scrolled down one line and the character immediately preceding the cursor will be cancelled. Data following the cancelled character will be shifted left one position for each deletion and word wrap will occur automatically if necessary.

If cancelling in the Compose Mode, width counting is continued and the values for the cancelled characters are restored to the deficit in the line measure. Backward cancelling beyond two line endings invokes the Edit Mode and width counting is disabled.

CANCEL WORD

Cancel Word, similar to Cancel Character, deletes all characters preceding the cursor, up to and including the first interword space encountered. If a word space is not present in the data line, the characters are cancelled up to the previous line ending (Return, Call Tab, Tab, Tab Return, End Line). Data following the cancelled characters will be shifted left and word wrap will be automatically performed where required.

If cancelling words in the Compose Mode, width counting is continued and the width values of cancelled data are restored to the deficit of the line measure. Cancelling beyond two previous line endings invokes the Edit Mode and width counting is disabled.

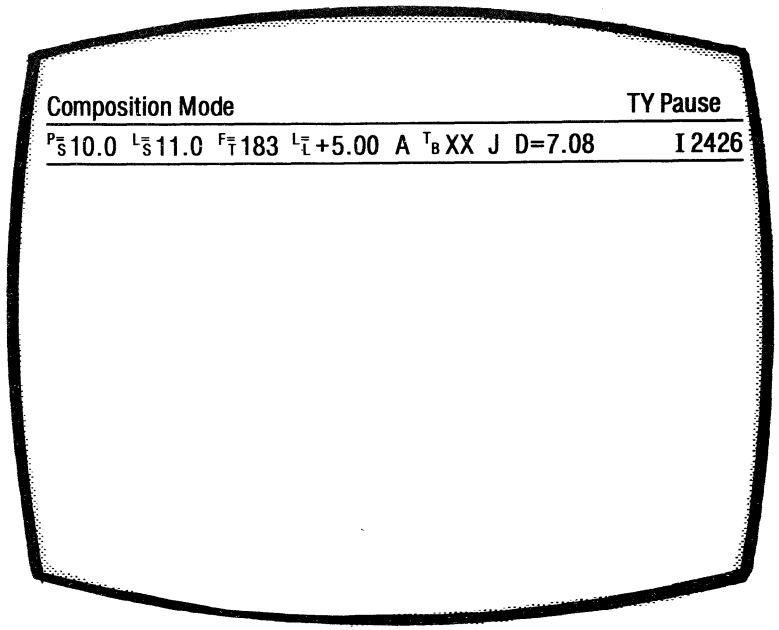
When cancelling a command string, Cancel Word will cancel characters back to and including the opening command symbol (<).

COMPOSITION MODE

Data can be entered in one of two modes: the *Compose Mode* or the *Edit Mode.* The *Edit Mode* is in effect at the start of each new job. The *Compose Mode* counts characters and character widths and is considered the standard mode of operation. It is enabled by keying *Compose* and *Execute.* "Composition Mode" flashes briefly on the top message line and a display of typesetting parameters appears on the second message line. These consist of Point Size, Line Space, Font, Line Length, Auto Line Ending, Tab Number, Quadding, Depth Count and Character Count. Default values are in effect for the first four at the start of each job. When parameter changes are entered through the keyboard, they are displayed at half-intensity on the bottom 13 screen lines. Text is displayed at full intensity. As data is entered, characters or function symbols are added at the position of the cursor.

While in the *Compose Mode,* character widths are counted during data entry. A line is justified when the accumulated width values exceed the line measure. The width count is updated by any forward or backward movement of the cursor. If the cursor is moved back beyond two previous line endings, including those that end tab columns, the *Edit Mode* is invoked. Character count continues but character widths are ignored until the *Compose Mode* is restored.

During file management procedures, typesetting parameters and the depth count are automatically extended to each new page. If a page is addressed out of sequence, parameters and depth count will not be extended. The Edit and Local Compose Modes will be in effect along with the default parameters.

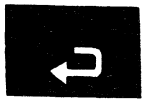


SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	

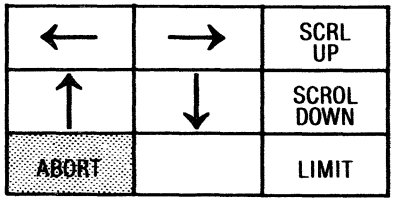
If default parameters are not applicable, new parameters can be inserted and the copy can be recomposed. Parameters can also be restored by addressing Page 1 and calling subsequent pages in sequential order. The operator should be aware that parameters continuity occurs only when pages are addressed sequentially. If a previous page is addressed without re-entering parameters, the default values will be in effect.

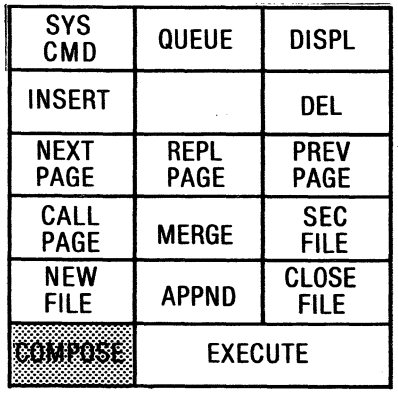
In the Compose Mode, the system defaults to the automatic line-ending mode. Character widths are counted to establish line breaks for hypnenation and justification. A half-intensity Return appears at the end of each justified line.

If a file is composed and an invalid command is detected, the Compose function is halted and the cursor will be positioned immediately after the error. An audible tone will sound and a message will be displayed in the status line. An erroneous command can be cancelled by pressing the Abort key on the Editing keypad prior to completing the command with Execute.



RETURN SYMBOL





Compose

The *Compose* function will justify text from the beginning of the current page up to but not including the character under the cursor. When in the Compose Mode, moving the cursor to the bottom of the page will compose the text line by line. The Compose Mode is changed to the Edit Mode by pressing the File Function key, followed by the letters EDIT, or the use of any editing function that causes the cursor to move back beyond any two previous line endings. If the cursor is moved while the Edit Mode is in effect, widths are not counted. After editing, data must be recomposed. Move the cursor to the bottom of the copy with *Limit/Scroll Up;* then press *Compose* and *Execute* to recount character widths and justify each line.

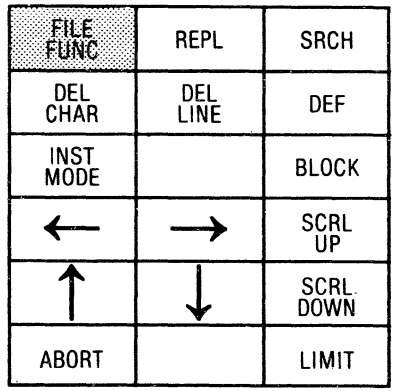
EDIT MODE

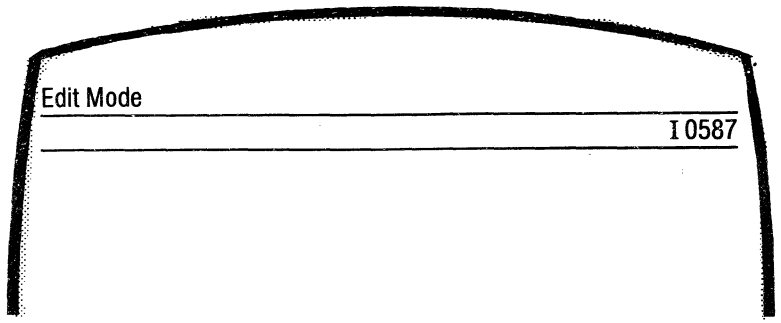
The Edit Mode is used when revising previously entered data by deleting or inserting characters. It replaces the Compose Mode whenever the cursor is moved in a reverse direction beyond two line endings and sounds an audible tone to alert the operator. It can be activated at the start of an editing session by pressing the File Function key and typing EDIT. It is indicated by "Edit Mode" on the top message line and by the absence of typesetting parameters on message line two. Characters are counted but character widths are ignored. If additional data is to be added to previously composed text, move the cursor to the proper location and press the Insert Mode key.

When editing has been completed, the page must be recomposed and replaced in the file. If a page is replaced in the Edit Mode, the Local Compose Mode will also be in effect at the start of the next page. Typesetting parameters will NOT be extended. The Edit and Local Compose Modes are also in effect if a page is addressed out of sequence, either forward or backward. New parameters can be inserted and will over-ride default values but no accurate record of accumulated depth will be available.

LOCAL COMPOSE MODE

The Local Compose Mode becomes effective after a file has been closed or when a file management procedure has interrupted the continuity of typesetting parameters. In the first instance, there is no corrective action necessary. In the course of addressing the next file, the Composition Mode will of necessity be restored. In the latter instance, with the Local Compose Mode in effect at the start of a new page, the beginning of the open file must be re-addressed and each page must be stepped through in the Compose Mode in order to restore lost parameters.

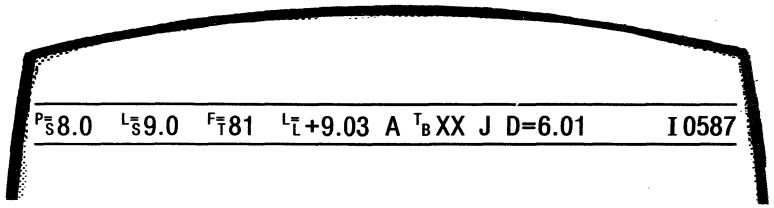




Procedure: *File Func*/EDIT/*Execute*

Insert Mode

The editing function *Insert* allows the operator to enter new data at the position of the cursor. The Insert Mode remains in effect until turned off by the alternate selection of the Insert key. The mode is indicated on status line two by the letter "I." There is no limit to the insertion. If more space is needed, the text stream will be opened automatically. Insert can be used in both the Compose and Edit Modes.



DEL CHAR	DEL LINE	DEF
INST MODE		BLOCK
←	→	SCRL UP

Delete Character

Individual characters or functions can be deleted by use of the cursor and the *Delete Character* key. Placing the cursor over the character and striking Delete Character (Del Char) will remove the character. All succeeding characters will move one position to the left. If the Delete Character key is held down, it will delete characters continuously until the key is released.

FILE FUNC	REPL	SRCH
DEL CHAR	DEL LINE	DEF
INST MODE		BLOCK

Delete Line

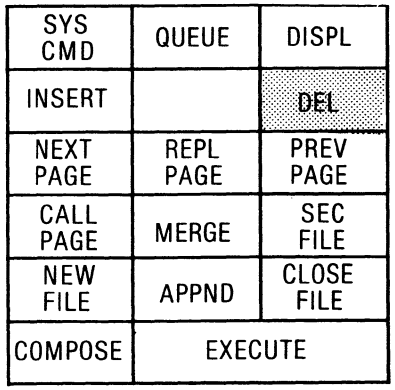
An entire line or part of a line can also be deleted. *Delete Line* erases all characters from the position of the cursor to the end of the screen line. As a precaution, the first depression of Delete Line (Del Line) will underscore all characters that will be deleted. Striking the key a second time will delete the line and word wrap and scrolling will occur to consolidate copy. If Delete Line is struck in error, the function can be cancelled by any cursor key or the Abort key.

FILE FUNC	REPL	SRCH
DEL CHAR	DEL LINE	DEF
INST MODE		BLOCK

DELETE PAGE

The current page or a page specified by number can be deleted from the primary file at any time. The page is deleted from the disk and subsequent page numbers are decreased by one to reflect the change. If the page is current, it is deleted from the disk but data is retained in memory and remains displayed on the screen. If appropriate, it can be transferred to become another page in the same file or in a secondary file. It can be cleared from memory by *Next Page/Execute* or *Delete/Display/Execute.*

If the number of the deleted page is greater than the current page, the current page number will not be affected. The Compose Mode and the progression of parameters for the following pages will be retained. If the deleted page preceded the current page, numbers for all succeeding pages will be reduced by one. The Edit and Local Compose Modes with default parameters will become effective.



TO DELETE THE CURRENT PAGE FROM THE DISK

Procedure: *Delete/Execute*

TO DELETE ANOTHER PAGE FROM THE DISK

Procedure: *Delete*/Page No./*Execute*

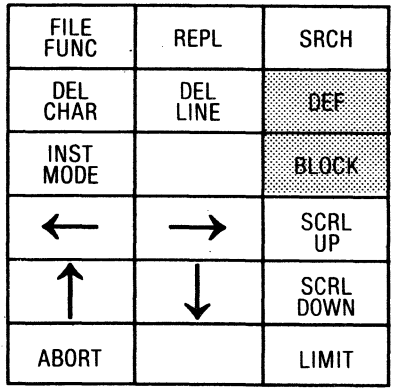
DEFINE/BLOCK

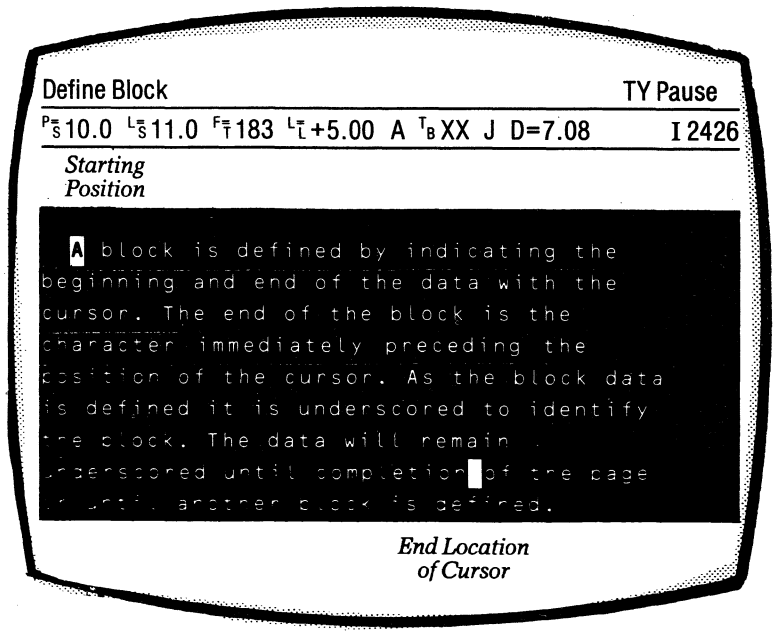
In editing, it is sometimes necessary to rearrange or delete sections of copy. To facilitate either of these functions, copy that starts at a point other than the beginning of the page can be isolated from the rest of the page by creating a *Block*. Place the cursor on the first character of the affected copy. Press *Define* and *Block*. Move the cursor horizontally or vertically to a position immediately after the last character of the block and press *Execute*.

As a block is being defined, an underscore appears below each character or line that the cursor passes through. Backward movement of the cursor erases the underscore. Data will remain underscored until the block is deleted or transferred. Disposition of a block should take place immediately after it has been defined. The function will be aborted should you attempt to define a second block. *Only one block can be in effect at any given time.* Once a block has been disposed of, another can be defined by repeating the procedure.

All cursor functions are enabled while defining a block. However, if the cursor is moved to a point which precedes the block, the function will be aborted. Use of a key other than a scrolling key also aborts the function.

Once defined, a block can be deleted, or transferred to another location in the same file or a secondary file. *Append* or *Insert* can be used to identify the destination point.




Define Block
TY Pause
P=S 10.0 L=S 11.0 F=T 183 L=L +5.00 A TB XX J D=7.08
I 2426
Starting Position
A block is defined by indicating the beginning and end of the data with the cursor. The end of the block is the character immediately preceding the position of the cursor. As the block data is defined it is underscored to identify the block. The data will remain underscored until completion of the page or until another block is defined.
End Location of Cursor

DEFINE BLOCK

Procedure: Place the cursor on the start of the block/*Define*/*Block*/ Move the cursor to the end of the block/*Execute*

DELETE/BLOCK

Once defined, all data in a block can be deleted. Word wrap and scrolling will occur automatically to consolidate the rest of the page. If a block is deleted while the Compose mode is in effect, data will be recomposed up to the cursor. The cursor will not be moved unless it has been positioned on a character within the block. It will be repositioned on the first character following the end of the deleted block. If the block is not on the screen during the deletion, completion will be indicated in the status area.

Procedure: *Define Block*/Move cursor to define block/*Execute/Delete/Block/Execute*

APPEND/BLOCK

If required, a defined block can be appended to either the primary or a secondary file and becomes the last page of the addressed file. The block is appended to a secondary file by supplying the name of the secondary file. (See Append).

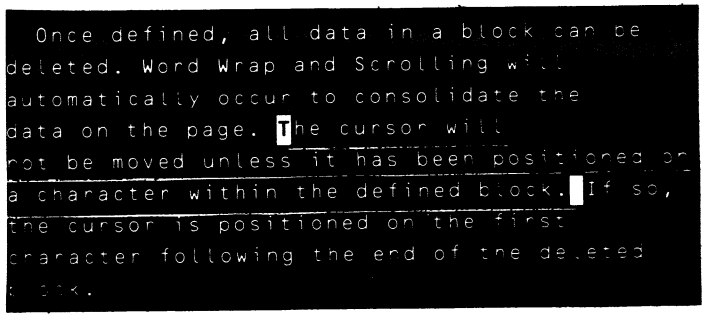
APPEND BLOCK, PRIMARY FILE

Procedure: *Define/Block*/Move cursor to define block/*Execute/Append/Block/Execute*

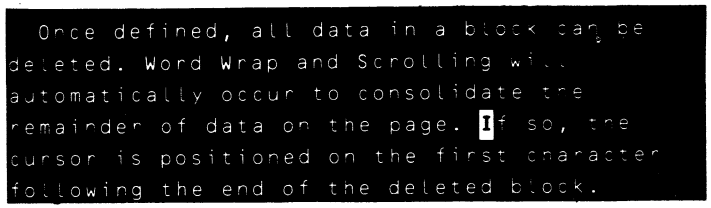
APPEND BLOCK, SECONDARY FILE

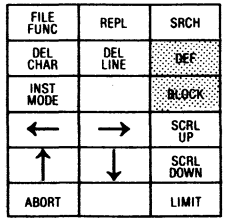
Procedure: *Define/Block*/Move cursor to define block/*Execute/Secondary/Append/Block*/File Name/*Execute*

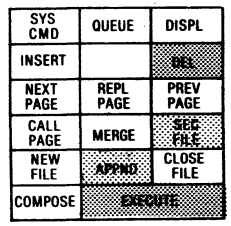
DELETE BLOCK


Once defined, all data in a block can be
deleted. Word Wrap and Scrolling will
automatically occur to consolidate the
data on the page. The cursor will
not be moved unless it has been positioned on
a character within the defined block. If so,
the cursor is positioned on the first
character following the end of the deleted
block.

BLOCK DELETED


Once defined, all data in a block can be
deleted. Word Wrap and Scrolling will
automatically occur to consolidate the
remainder of data on the page. If so, the
cursor is positioned on the first character
following the end of the deleted block.





SEARCH

Once data has been called to the screen, the operator can address a specific area almost instantly by pressing *Search* and typing up to 16 characters to identify the objective, followed by *Execute.* A typical objective would be the first correction point. The Search is conducted from the cursor location to the end of the page. When a match is found, the cursor will stop on the first letter. If the search string appears more than once on a page, repeated depressions of *Execute* will advance the cursor to the next occurrence. If the cursor is moved or if any corrections are made, *Search* and *Execute* must both be keyed to continue the search.

Search can be redefined at any point by repeating the initial procedure and identifying the new objective. If the search string does not appear on the page, the message "No Match" will be displayed in the status area. The cursor will remain on the last data string found in the search routine.

RANDOM DATA SEARCH

Variable data can be addressed by substituting a question mark (?) in place of unknown data. The question mark represents missing characters associated with the data string. For example, the data string **?????graphic** would locate the following combinations: Compugraphic, COMPUGRAPHIC and typographic since the search string consists of *graphic* preceded by any five characters, including word spaces.

When the question mark is used as true punctuation within a data string, it is differentiated from the variable symbol by entering a colon immediately preceding the question mark. A search for "Where?" would be entered as: **"Where:?"**. This also applies to a colon if used as punctuation within a search string (::).

Search Data String [Where?]

Where:?

Search Data String [Follows:]

Follows::

Variable Data String [?????graphic] matches:

Compugraphic
COMPUGRAPHIC
CompuGraphic
typographic

SEARCH ONLY

New Procedures:

Search / Enter Data / Execute

Default Procedures:

Search / Execute

Repeat Search Procedures:

Search / Execute / Execute / Etc.

SEARCH AND REPLACE

When the same correction is to be made several times in a file, the *Search* and *Replace* routine can be used effectively. The system is instructed to find the original version of data and replace it with the corrected version. For example, the day and date of an event can be changed throughout a file by pressing *Search;* typing the original schedule; then pressing *Replace,* followed by the new schedule and *Execute*:

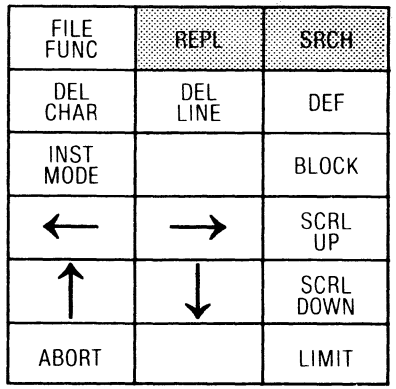
Search Key:*Monday, May 2* **Rep:***Wednesday, May 4*

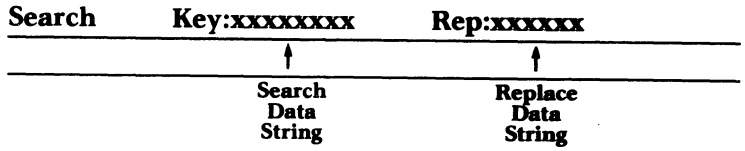
Each depression of *Execute* addresses the next occurrence of the Search string and substitutes the Replace string.

As data is entered, it appears on the top message line. Any displayable text character or function symbol can be entered in the Search string, including the question mark for variable data.

For operator convenience, the search routine is case-sensitive. A search for lower case characters will locate both lower case and upper case characters. For example, when searching for the word *system,* the program will also recognize *System* and *SYSTEM.* Conversely, however, if upper case characters are defined, the program will find only upper case. It will not recognize lower case characters. Once defined, the search string will be compared with data in the page. When a match is found, the cursor will be positioned on the first character of the string. Where possible, the searched data will be moved vertically to the center of the screen.

Once entered, the *Search/Replace* strings remain active until one or both are redefined. Errors made while defining a string can be cancelled with Cancel Character or Cancel Word.





LOWER CASE SEARCH MATCHES

Compugraphic
compugraphic
COMPUGRAPHIC

UPPER CASE SEARCH MATCHES

COMPUGRAPHIC

Defining Replace Data String

The Replace data string, like Search, can be up to 16 characters. The Replace string *can include functions, Spacebands (Interword Spaces), Hyphens and Returns,* and will be displayed on the message line. However, no variable character delimiters (:?) can be included in the Replace data string. *The data strings are entered by the sequential entry of Search/Replace or Replaced after the Search key has been found.* When defining the Search and Replace strings, the Cursor Right and Cursor Left functions can be used to edit the data. Once entered, the Replace data is defaulted and remains in effect until entry of a new Replace string or being cancelled. If the data cannot be found, the message "No Match" will be displayed in the message area. The cursor is then moved by the operator to the bottom of the page and the page replaced.

During replacement, if the terminal memory limitation of 6,000 characters is exceeded, it will be indicated with the status message "FULL." Only that portion of the Replace data string that can fit is entered. The page must be divided and stored by the operator with the "Insert," function on the File Management keypad (See File Management, Page 28).

Search Key:xxxxxxxx Rep:xxxxxx

SEARCH AND REPLACE

New Procedures:

Search/Enter Data/Replace/Enter Data/Execute

Default Procedures:

Search / Replace / Execute

Search / Enter Search Data / Replace / Execute
(New Search String)

Search / Replace / Enter Replace Data / Execute
(New Replace String)

Repeat Search Procedures:

Search / Replace / Execute / Execute / Etc.

Search Key:xxxxxxxx Rep:xxxxxx

FULL

Search and Delete

In addition to replacement, a Search string can be deleted. *Entering a blank Replace string in place of data will delete the Search string when it is found.*

Operator Defined Search and Replace

When the replacement of the data string varies, the data can be replaced at the discretion of the operator. Executing Search *without* Replace, positions the cursor on the next occurrence of the Search string. The Search string can now be Replaced with the default data or by entering a new Replace data string. When entered, the Search and Replace routine can be automatically continued by the repeated operation of Execute. If the replacement of data is not required, the functions Search/Execute continues the Search.

File Search and Replace

The Search and Replace routine can be utiltized through an entire primary file. File Search and Replace is executed with the *File Function* key located on the editing keypad. When the File Function is activated, it is followed by entering the Search and Replace data string . As each page is completed or if the Search string is not found on the current page, the page is automatically replaced in the file. The next page will then be entered onto the videoscreen with the Search starting at the first character in the new page. The Search and Replace is continued for each occurrence of the string by striking *Execute* until the end of the file. If the Search is in the Compose Mode, the entire file will be recomposed as it is being replaced.

SEARCH AND DELETE

Procedures:

Search / Enter Data / Replace / Execute

DISCRETIONARY SEARCH AND REPLACE

New Procedures:

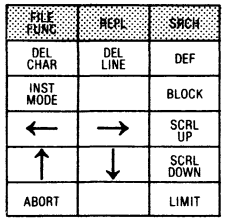
Search / Execute
Replace / Enter Data / Execute

Default Procedures:

Search / Execute
Replace / Execute

Repeat Search Procedures:

Search / Replace / Execute / Execute / Etc.



FILE SEARCH AND REPLACE

New Procedures:

File Function / Search / Enter Data / Replace / Enter Data / Execute

FILE SEARCH ONLY

Default Procedures:

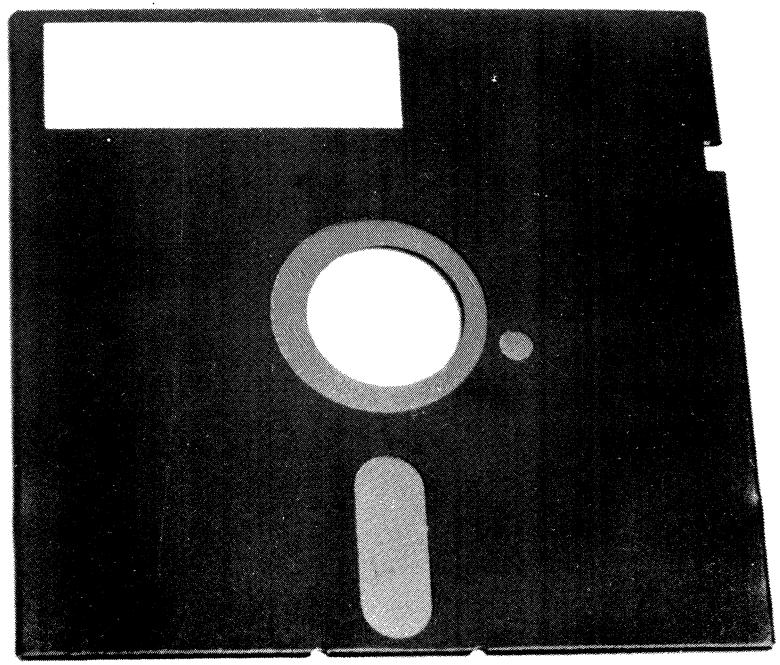
File Function / Search / Enter Data / Execute

SEARCH/REPLACE ERROR MESSAGES

Message	Reason
NO KEY	Search function executed without a defined Search key.
FIELD FULL	Too many characters entered for key data.
NO MATCH	The defined Search Key has not been found.
FULL	Exceeded memory limitation of 6,000 characters. Use file management function "Insert" to divide page.
INVALID INPUT	Erroneous character entered in Search key.
COMPOSE ERROR	Composition error made during input. Search will not be continued until corrected.
DISK ERROR	Read/Write disk error.

File Management

File management is the organization and editing of file data on the job disk. File Management consists of writing, reading, deleting, replacing, updating, merging and storing new files on the mini diskette. All File Management is accomplished via the mini diskette in conjunction with the functions on the Left and Right Control Keypads. The files are divided into pages with a limit of 6,000 characters per page, the maximum storage capacity of the terminal memory. During data input, when a count of 5500 characters is obtained, it is indicated by an audio tone. The tone will be repeated sequentially with additional data entry. When memory limitation of 6000 characters occurs operation is inhibited, indicated by the message "Page Full." *It should be noted that the diskette is limited to MCS operation.*



MINI DISKETTE

File Name

A file is identified by an operator designated file name. The file name can consist of one to twelve (12) alphanumeric characters including hyphens and floating accents.

Composable Text File (TC)

Designated as "TC," a text file is the user or job file created by the operator. It consists of alphanumeric text (characters) and typesetting parameters and functions.

Font Attribute File (FA)

"FA" indicates that a designated Font File with attributes has been loaded into terminal memory.

Font Complement

"FC" indicates the character complement of the Font File loaded into terminal memory.

PRIMARY MODE

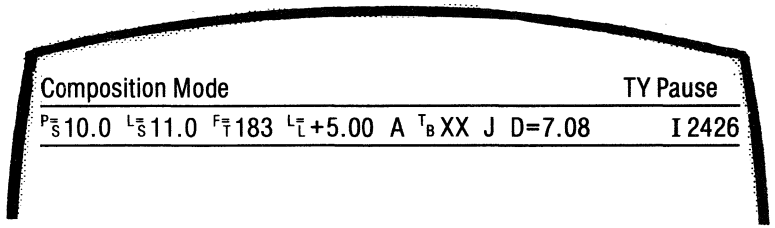
There are two modes in which files can be addressed: Primary and Secondary. The Primary Mode is used when a new job is to be input or an existing job is to be edited. It is established automatically by the manner in which the job is addressed. When *New File*/file name/*Execute* are used to start a new job, the Primary Mode is enabled. When an existing job is to be corrected, *Call Page*/file name/*Execute* address the job in the Primary Mode. After corrections are made, the page is replaced on the disk and overwrites the original page. These procedures can be performed only in the Primary Mode, which is indicated by the letter "P" on the top message line, to the left of the disk drive area.

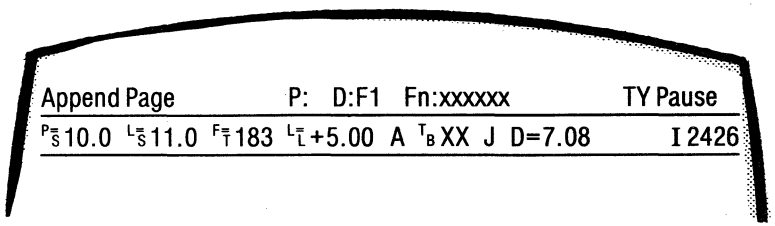
SECONDARY MODE

The Secondary Mode establishes communication between two files and enables transfer of data from one file to the other. It is activated by the *Secondary File* key on the file management keypad. It is used prior to the function involved and is indicated by the letter "S" on the top message line. The system can be addressed to one file in the Primary Mode and another file in a Secondary Mode. A series of files can be addressed in turn in the Secondary mode.

An entire page or a block of copy can be transferred from the Primary file to a Secondary file via Append or Insert. A page can be transferred from a Secondary file to the current page of a Primary file via Merge.

Functions that are applicable in the Secondary Mode include: *Append Page, Append Block, Call Page, Insert Page, Insert Block, Next Page* and *Previous Page.* All other file management functions must be performed in the Primary Mode.





PRIMARY FILE

Procedure: *New File/(file name)/Execute*
Call File/(file name)/Execute

TO TRANSFER DATA TO THE PRIMARY FILE

Procedure: *Sec File/Merge/(file name)/Page No./Execute*
Sec File/Merge/Next Page/Execute

TO TRANSFER DATA WITHIN THE PRIMARY FILE

Procedure: *Append/Execute* or *Append/Block/Execute*
Insert/Page no./Execute or *Insert/Block/Page no./Execute*

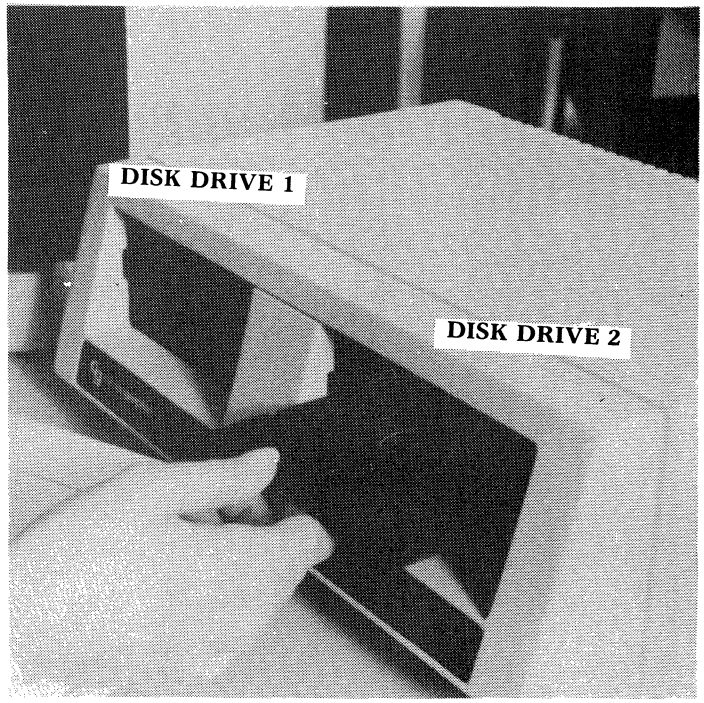
TO TRANSFER DATA FROM THE PRIMARY FILE

Procedure: *Sec File/Append/(file name)/Execute* or
Sec File/Append/Block/(file name)/Execute
Sec File/Insert/(file name)/Page no./Execute or
Sec File/Insert Block/(file name)/Page No./Execute

DUAL DISK OPTION

As a standard system, the **MCS** consists of one terminal that is used in conjunction with a single disk drive. In single disk drive operation, only one disk can be engaged at a time. The dual disk drive *option* permits the operation of an additional drive which allows alternate communications between two disks. The disk drives are identified as **F1** (Floppy Disk Drive 1) and **F2** (Floppy Disk Drive 2). The left Disk Drive (F1) is the default drive and is in effect on start up. The right Disk Drive (F2), is activated by the operator entering **F2** in the disk drive data field in the message line. The system remains addressed to Drive 2 until Drive 1 is selected. In addition to the drive number, the operator must also identify the file name and/or page number where required. The selection of disk drives also applies to file management applications within Secondary files.

Dual Disk operation doubles both the storage and retrieval capacity of file data. Repetitive data or formats can be copied and merged from one disk to the other. (See Dual Work Station and Multi Disk Reference Manual, Part No. 214811).



EXECUTE

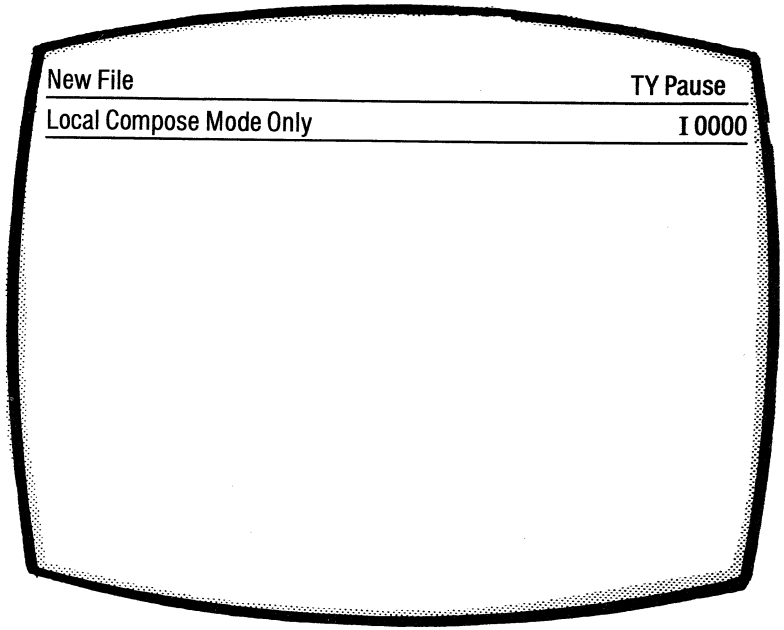
Execute is used immediately after each File Management function to activate the command. In some cases the file name or other pertinent data must be supplied by the operator prior to pressing *Execute.* All related functions can be aborted prior to their completion.

Execute is also used to produce the close Control symbol when supplying data for a programmed key.

NEW FILE

The *New File* function is used to open a new file and establish default parameters for data entry. The defaults consist of the Local Compose Mode and typesetting parameters for Point Size (8.0), Default Font, Line Space (8.0), Line Length (15.00), Auto Line Ending (A), Tab Number (xx), Quadding (J), Depth and Character Count per page. When *New File* is enabled, the system responds by requesting the file name. Up to twelve alphanumeric characters can be entered for the name. The function is completed with *Execute.* The cursor is positioned at the top of the page and the system is addressed to Page 01. When the *New File* function is completed, new parameters should be entered and file data can be input to the terminal memory and videoscreen. To activate character counting and justification, press *Compose/Execute* prior to entering text material.

Procedure: *New File*/(file name, ← drive no.)/*Execute.*



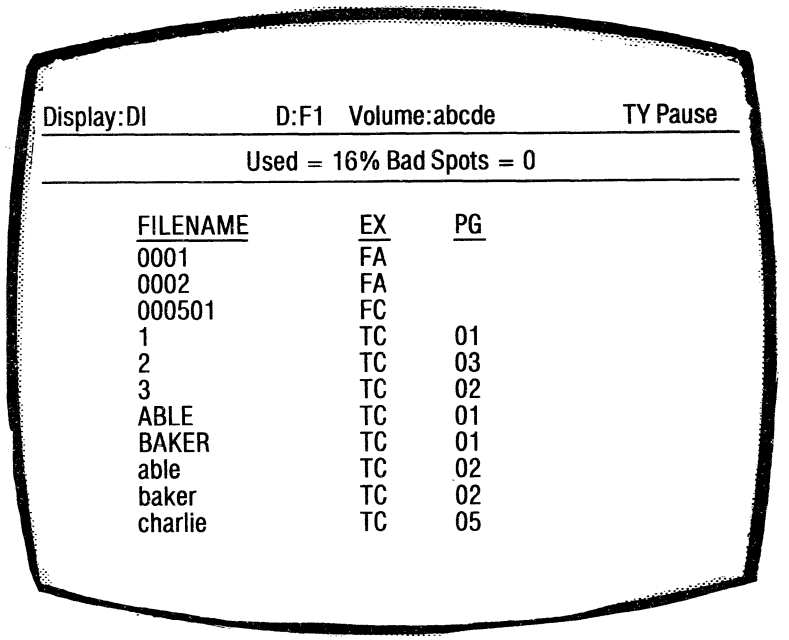
SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	

DISPLAY INDEX

A directory displaying the file names stored on the disk can be viewed at any time. The Display function followed by the letters "DI", the Disk Drive identification (F2, H1, etc.) and *Execute* will call the directory to the screen. The file names, number of pages for text files and the user disk volume name will also be listed. Numeric file names are listed first and are followed by names that start with upper case letters, then those that start with lower case letters. Assigning file names with that priority in mind can be helpful in locating files.

The file category is identified by TC for text files, FL for font dress files, FA for font attribute files and FC for font complement files. The percentage of the disk that is in use and a possible error factor appear in the status line. If the directory is called up while a new job is being entered, it will not be updated until the current file is closed. The new file name will be listed but the number of pages will be 00.

Scroll Up and Scroll Down can be used to display file names beyond the eleven that can be displayed at one time. While the index is being displayed, files may be deleted and jobs can be entered in various queues. However, the index will not update until it is cleared and addressed again. If a time-out occurs while the index is on the screen, pressing any key will restore the display.



ABBREVIATIONS:
EX - Extension
PG - Page
TC - Composable Text File
FA - Font Attribute File
FC - Font Complement File

SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	

DISPLAY INDEX

Procedure: *Display*/DI/Disk Drive No.(F1-4)/*Execute/Scroll Up/Scroll Down*

CLEAR INDEX

Procedure: *Abort*

APPEND PAGE

As data is input it is entered as the current page in terminal memory. When completed or when the character count limit is being approached, the page should be stored on the disk with the *Append* function. All text, *up to the cursor,* will be stored. Data transferred to the disk is cleared from both the videoscreen and terminal memory. Each new page is entered sequentially. Page numbers are indicated in the status area. *To ensure justification, the last line of each page must have a line-ending furnished by the operator rather than one supplied by the justification program.* The Compose Mode must also be in effect.

If the system is addressed to the last page of a file, the page number will be incremented. If addressed to any other page, the page number will remain constant. Data will be appended to the end of the file.

If only a portion of a page is appended, data remaining on the screen becomes the start of the succeeding page. Data can be appended to the open file or to any file addressed in a Secondary mode.

APPENDING PAGE TO THE END OF A NEW PRIMARY FILE

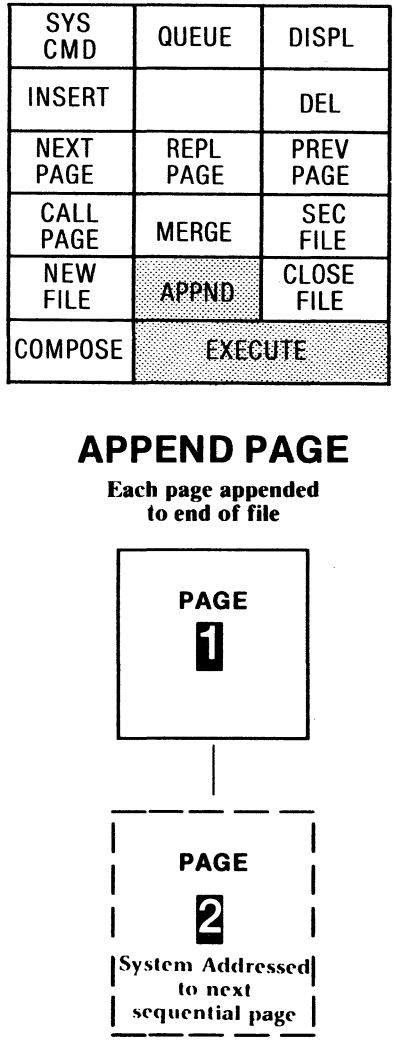
Procedure: *Append/Execute* will store the page at the end of the file; system will be addressed to the next sequential page.

APPEND PAGE TO THE END OF AN EXISTING PRIMARY FILE

Procedure: *Append/Execute* will enter the page at the end of the file; system remains addressed to the original page.

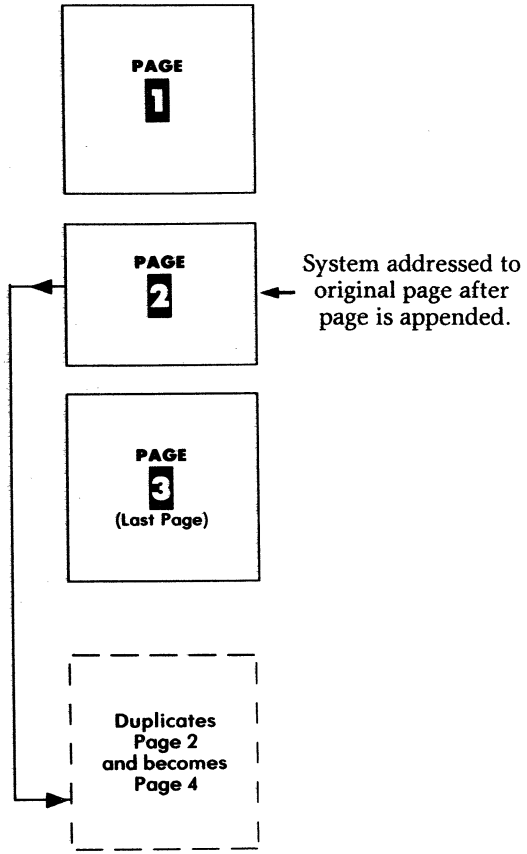
APPEND PAGE TO THE END OF ANOTHER FILE

Procedure: *Sec File/Append/*(file name)*/Execute;* system remains addressed to the original page.



APPEND PAGE TO END OF CURRENT FILE

(Append Page 2 to end of file)

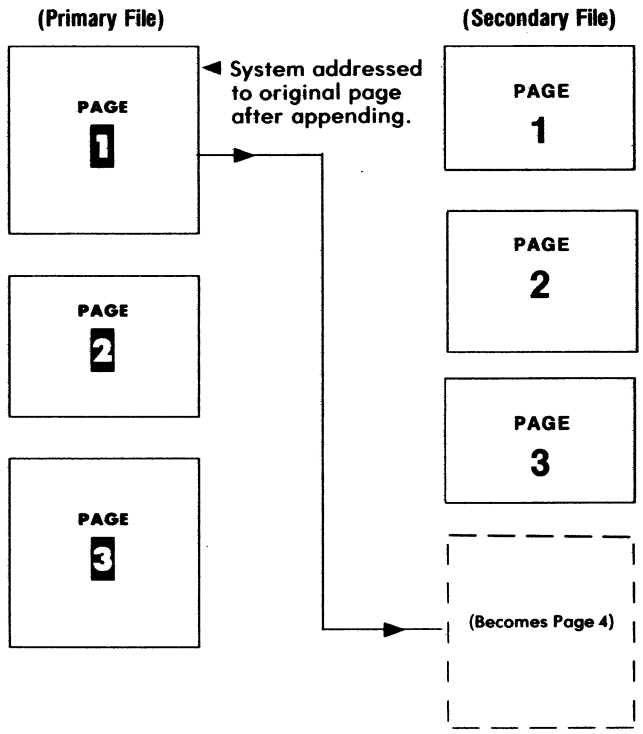

PAGE 1
PAGE 2
System addressed to original page after page is appended.
PAGE 3
(Last Page)
Duplicates Page 2 and becomes Page 4

APPEND PAGE TO END OF CURRENT FILE

Procedure: *Limit/Scroll Up* to position cursor at the bottom of the data. *Append/Execute* will enter the page at the end of the file and the system will be addressed to the original page.

APPEND PAGE TO ANOTHER FILE

(Add Page 1 to end of another file)



APPEND PAGE TO END OF ANOTHER FILE

Procedure: Move the cursor to the end of data; *Secondary*/file name/*Append/Execute*

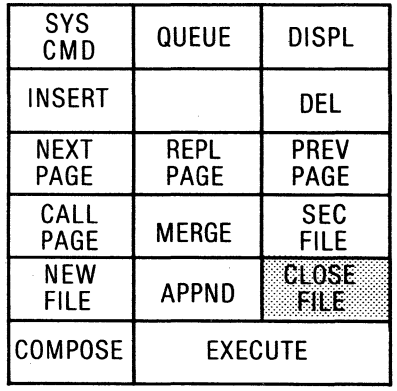
CLOSE FILE

An open file is established in two ways: by *New File* to enter new copy into the system and by *Call Page* to edit an existing file. Both methods invoke the Primary mode of operation. New data can be keyboarded and pages can be appended to the file. For editing, pages can be corrected and replaced on the disk. Since only one file can be open at a time, when input or editing has been completed, the file must be closed. This is accomplished by the *Close File* and *Execute* keys on the File Management keypad. The operator should append or replace the current page before closing a file. The Edit and Local Compose Modes become effective after a file has been closed. The file may now be typeset or entered into the queue of any peripheral device. Another file can then be opened.

If the current page was not appended before the file was closed, data will remain in the terminal memory and will be displayed on the screen. The page can be added to the file by addressing the previous file in a secondary mode and appending the page: *Secondary/Append/*File Name*/Execute.*

If a disk is inadvertently removed from a disk drive while a file on that disk is open, an audible tone will sound and the message "File Open" will appear on the second message line. The disk should be replaced immediately and the status of the file examined. In some cases it may first be necessary to append or replace the page on the screen. Most important, *Close File* and *Execute* must be keyed before the disk is finally removed. If an "open file" condition is permitted to continue, it will not be possible to address any file in the Primary mode and the system will have to be reprogrammed.

Procedure: *Close File/Execute*



CALL PAGE

To Open a File

A two-fold function, *Call Page* is applicable to recalling a file or an individual page. To open a file, pressing *Call Page* positions a cursor on the File Name (Fn:) in the status line. Entering the file name followed by *Execute* opens the file and transfers the first page of the file from the disk to the screen. The cursor will be in the home position.

To Address a Page

Once a file has been opened, *Call Page* positions the cursor on the current page number (Pg:) in the status line. If no page number is entered, *Execute* erases all data in memory and causes the current page to be called to the screen. If a page number is supplied, that page will be recalled. If a page is addressed out of sequence, the Local Compose Mode and default parameters will be in effect. Wherever possible, the Compose Mode should be preserved by addressing pages in numerical sequence. *Each page should also be ended with a full-intensity manual Return.*

TO OPEN AN EXISTING FILE

Procedure: *Call Page*/File name/Disk drive/*Execute.*

CALL FILE AND PAGE NUMBER

Procedure: *Call Page*/File name/Page number/
Disk drive/*Execute.*

CALL PAGE, SECONDARY FILE

Procedure: *Secondary/Call Page*/File name/Disk drive/
Page number/*Execute*

SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	

PROTECTED FILES

A file is assigned temporary "protected" status when it is resident in a queue or when it is in use on another terminal. The file cannot be addressed in the Primary Mode until the file is closed on the other terminal or until it clears the queue. The message line will display "Protected" while that status remains in effect.

If a power failure occurs while a file is open, that file becomes permanently "protected". However, a copy can be made of a permanently protected file. The copy will have the same number of pages and can be treated as though it were the original file. If copied on the same disk, a different name must be assigned.

TO COPY A PERMANENTLY PROTECTED FILE

Procedure: *System Command*/C O P F/Drive number/Old file name/
Drive number/New file name/*Execute*

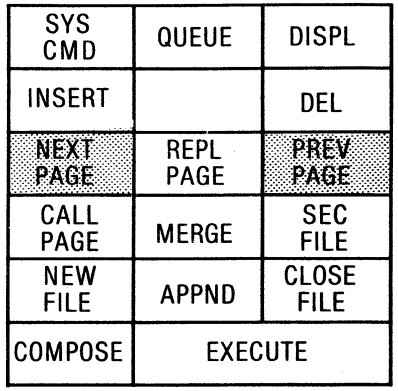
NEXT PAGE

This function is often used when editing copy where corrections occur at scattered intervals in the file. Some pages will not require editing. Once a file has been opened, pages should be addressed in sequence. In order to update typesetting parameters and maintain an accurate depth count, the Compose Mode must be in effect at all times. If changes have been made to a page, *Replace Page* and *Execute* cause the entire page to be recomposed and transfer it to the disk. The next page is also called to the screen. If no corrections are to be made in the page, *Next Page* and *Execute* cause it to be recomposed. Parameters are updated, but no data is transferred to the disk. The screen is cleared and the next page is displayed. Parameters in effect at the end of the previous page remain in effect.

If even a single character is changed, *Next Page* should not be used since it will invoke the Edit and Local Compose Modes. All edited pages must be ended with *Replace Page* and *Execute.* This function can also be used with Merge.

PREVIOUS PAGE

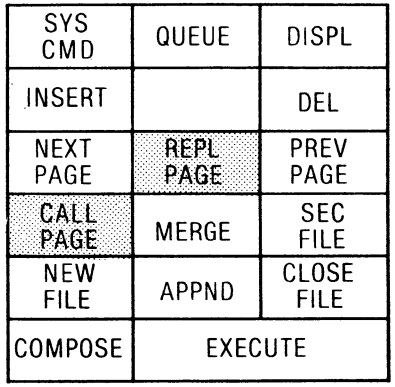
Once a file has been addressed, pages can also be accessed in reverse sequential order. The functions *Previous Page* and *Execute* clear the current memory, reduce the page number by one and display the previous page. If a page is called in reverse order, the Edit and Local Compose Modes and default parameters become effective. If Page 1 is addressed in this manner, all necessary parameters appear in that page and valid updates can be made by recomposing. If a page other than Page 1 is accessed, it may be necessary to redefine parameters and other typesetting functions. The depth count defaults to zero if a page is addressed out of sequence.



REPLACE PAGE

When a page has been recalled and edited, it must be replaced in the Compose Mode in order to record changes on the disk and extend current parameters to the next page. In the Compose Mode, *Replace Page/Execute* automatically positions the cursor at the end of the page, composes the page and overwrites the unedited version on the disk. It also calls the next page to the screen. The cursor will be on the first character of the page and all parameters in effect at the end of the previous page will remain in effect. The page number and character count are updated and reflect the status of the new page.

If a page is replaced while in the Edit Mode, the Local Compose Mode with default parameters will be in effect for the subsequent page. This requires additional operator input to correct the situation and should be avoided. In order to update the Depth Count and restore parameters, you must address Page 1 and advance to the point of the error in the Compose Mode.



REPLACE PAGE

Procedure: *Call File/*(type file name)*/Execute/*Make corrections*/Replace Page/Execute.* (Where necessary, *Compose/Execute* before *Replace Page.*)

INSERT PAGE

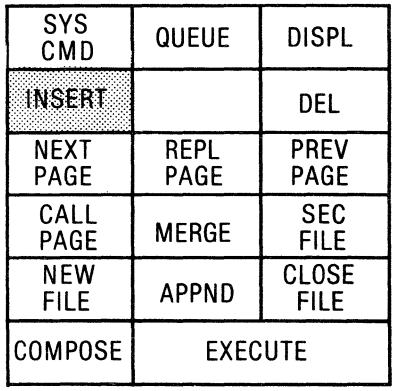
A page or any portion thereof can be transferred from the screen and inserted as another page in a file. The destination file can be either Primary or Secondary. Data in memory, from the beginning of the page up to the cursor, is transferred to the specified file and is deleted from the screen. Page numbers of the destination file will be incremented.

It should be noted that the inserted data is a duplicate of the original and if not required in both locations, it should be deleted from the original page. Replacing the page serves that purpose. If data is to be retained in both locations, do not replace the page and the original version will remain intact on the disk. Be sure that all necessary parameters are present after any transfer of data.

The Insert and Block functions can also be combined to insert data that starts at a point other than the beginning of the original page.

MEMORY FULL

Although data can be transferred to various locations with *Insert,* its primary application occurs during editing when the screen memory of 6,000 characters has been exceeded. A Memory Full message is displayed and typewriter keys are disabled. *Insert* enables the operator to divide the page in two. Move the cursor back to a previous paragraph ending or quadded line. Press *Insert/Execute.* Copy preceding the cursor will be deleted from the screen and will be inserted ahead of the current page. When editing has been finished, *Replace Page/Execute* completes the procedure. Any subsequent page numbers will be incremented by one. A page can be divided into a series of pages by repeated use of the Insert function.



MEMORY FULL/DIVIDE PAGE DURING EDITING

Procedure: Move cursor to logical break point/
Insert/Execute/
Compose/Execute/Replace Page/Execute

INSERT PAGE WITHIN PRIMARY FILE

Procedure: Move cursor to the end of current page/
Insert/Page No./*Execute*

INSERT PAGE—PRIMARY TO SECONDARY FILE

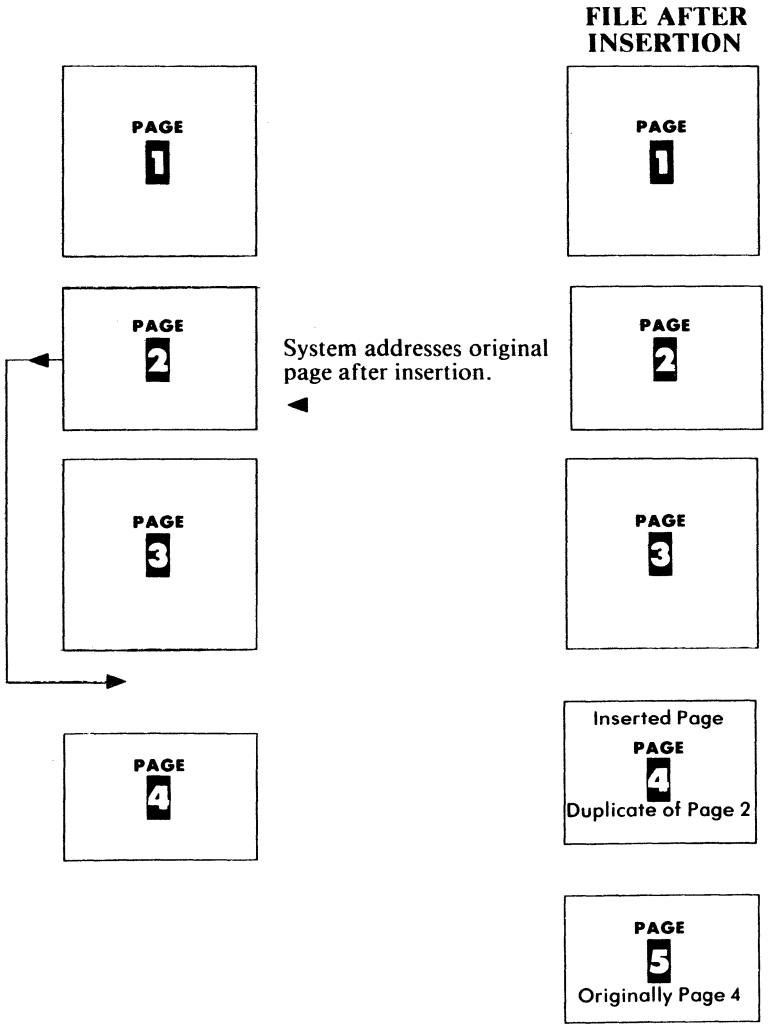
Procedure: Move cursor to end of current page/
Secondary/Insert/File Name/Page No./*Execute*

INSERT BLOCK

Procedure: (Block must have been defined previously)
Insert/Block/Page No./*Execute;* or
Secondary/Insert/Block/File Name/Page No./*Execute*

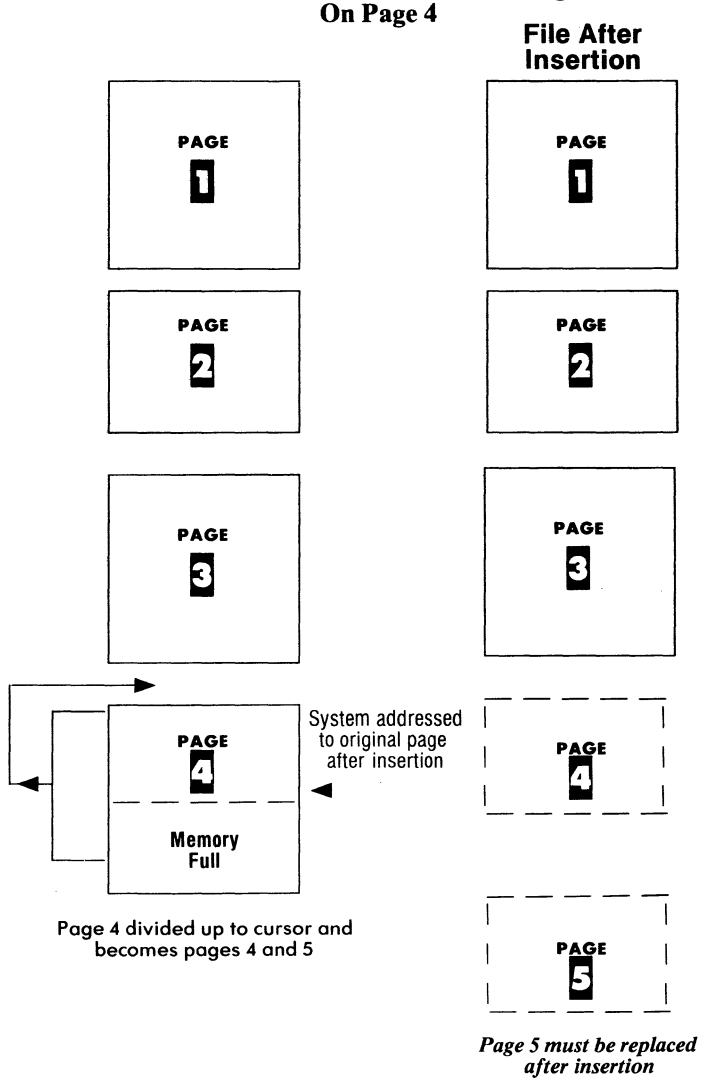
INSERT ONE PAGE AHEAD OF ANOTHER

(Insert Page 2 ahead of Page 4)



DIVIDING PAGE WITH INSERT

Example, Memory Full Occuring On Page 4



Page 4 divided up to cursor and becomes pages 4 and 5

Page 5 must be replaced after insertion

INSERT BLOCK

A segment of data can be defined as a block and inserted as *a page within a specified file.* The file can be either Primary or Secondary, and when the block is inserted, page numbers will be automatically incremented. Once the defined data in memory is transferred, it will be deleted from the screen. If the transferred data contained parameters they will no longer be in effect for the remaining data if recomposed.

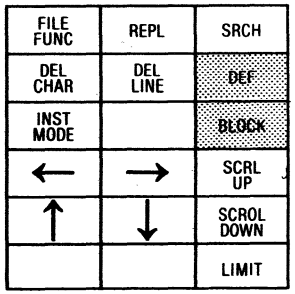
If the inserted block includes parameters, they will be in effect when the newly created page is composed.

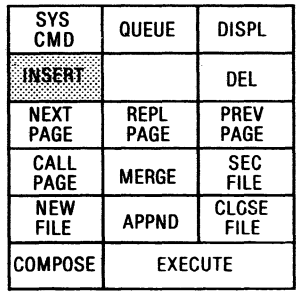
INSERT BLOCK

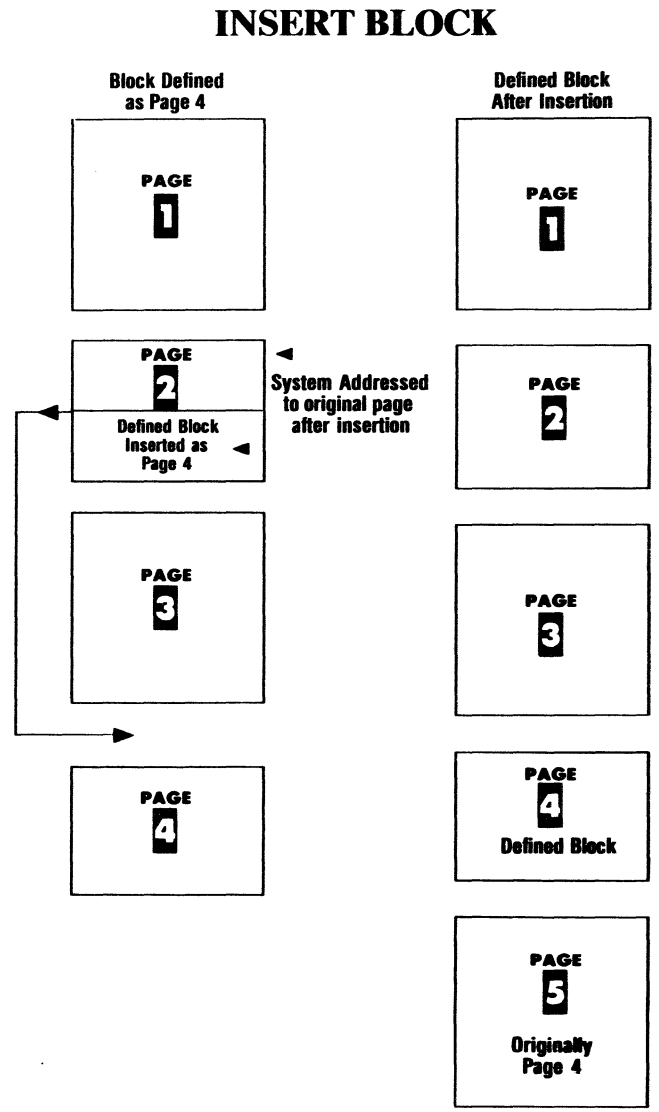
Procedure: *Define Block/Insert/Block/*Page No./*Execute*

INSERT BLOCK (Secondary File)

Procedure: *Define Block/Secondary/Insert/Block/*File Name/Page No./*Execute*



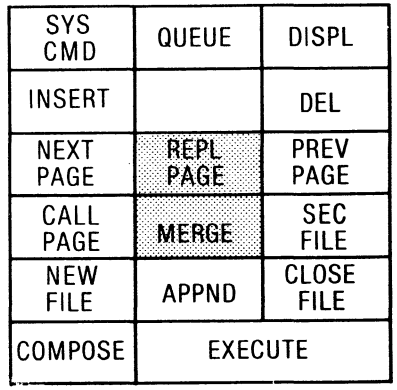




MERGE

A typical application of this function is to combine two pages in the same file into a single page. Total characters must not exceed 6000. Address the first of the two pages and place the cursor at the point where the second page is to be added. *Press Merge/*page no./*Execute.* A copy of the latter page will be added to the current page. If the Compose Mode is active, the new data will be composed on input. The cursor will appear at the end of the new copy. The page can then be replaced on the disk. Since the merged data is duplicated in the file, it may be necessary to delete it by its original page number or by calling up the page and pressing *Delete/Execute.*

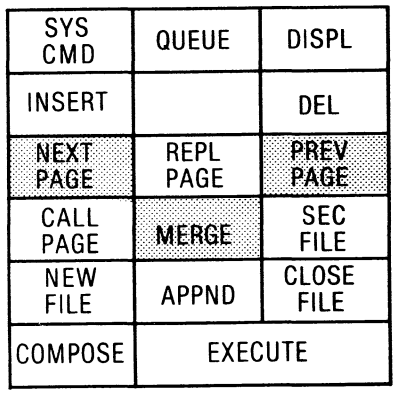
If data to be merged is in a different file, the procedure differs slightly. After the cursor has been positioned, Press *Secondary/Merge/*file name/page no./*Execute.* If the file is on another disk, the drive must also be identified.



MERGE/NEXT PAGE

Next Page can be used in place of the page number to combine consecutive pages in the Primary file. Address the first of the two pages, place the cursor at the point of insertion and press *Merge/Next Page/Execute.* Data is added at the cursor location. Next Page does not increment in the Primary file. If Page 1 is the current page, Next Page addresses Page 2 only.

A series of pages can also be transferred to the Primary file from a Secondary file. Once the first page has been transferred, Next Page can be repeated to access subesequent pages in that file. *Secondary/Merge/Next Page/Execute* will access the page with the next higher number than the one which appears in the message line. If "Pg:02" appears, Next Page will access page 3. Unlike the Primary file, the page number in the Secondary file does increment.



MERGE/PREVIOUS PAGE

This function can also be used to combine consecutive pages in the Primary file and is used instead of the page number. It applies when addressed to the second of two pages. Move the cursor to the point where copy is to be added and press *Merge/Previous Page/Execute.* New data will be added at the cursor location. Within the Primary file, Previous Page cannot be used to address a series of pages in reverse order. If Page 5 is the current page, Previous Page will always address Page 4.

A series of pages can be transferred from a Secondary file in reverse order. Once the first page has been transferred, Previous Page can be used repeatedly to access prior pages in that file. *Secondary/Merge/Previous Page/Execute* will transfer the page with the next lower number than the one that appears in the message line. The page number in a Secondary file does decrement.

ADD DESIGNATED PAGE TO BOTTOM OF CURRENT PAGE OF PRIMARY FILE

Procedure: Move cursor to the bottom of the current page
Merge/page no./*Execute*

ADD DESIGNATED PAGE TO TOP OF CURRENT PAGE

Procedure: Move cursor to top of current page:
Merge/page no./*Execute*

MOVE PAGE FROM SECONDARY FILE TO BOTTOM OF CURRENT PAGE

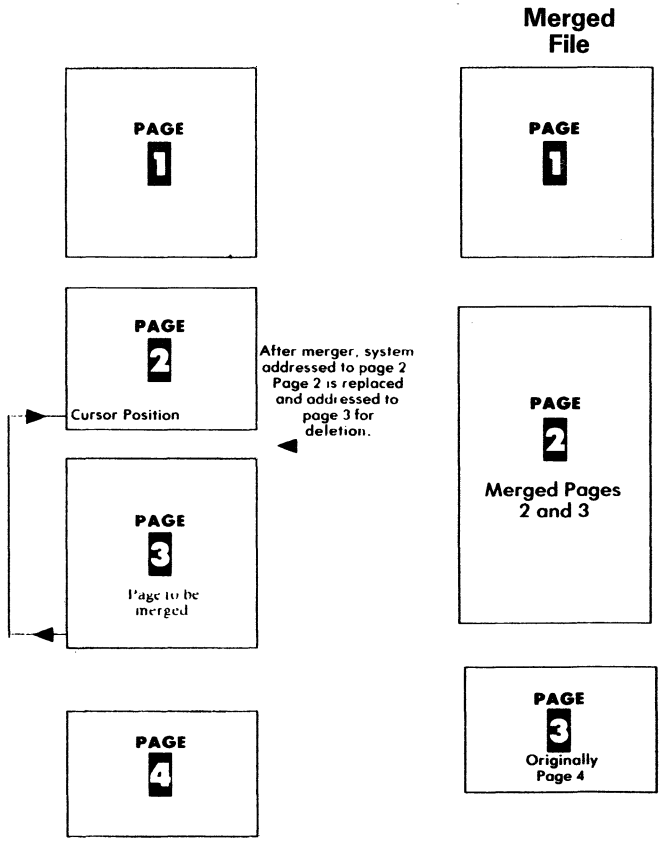
Procedure: Move cursor to bottom of current page:
Secondary/Merge/file name/page no./*Execute*

TO ELIMINATE MERGE OVERFLOW CONDITION

Procedure: Move cursor back to hard return:
Compose/Execute/Insert/Execute;
Limit/Scroll Up/Merge/Merge/Execute;
Replace File/Execute

MERGING PAGES

(Merging Page 3 with Page 2)



Preparing Copy For The Typesetter

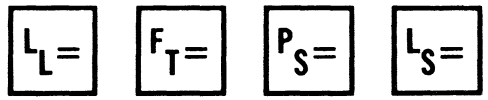
The primary objective of terminal procedures is to produce typeset copy. The terminal keyboard provides access to all photographable characters on the associated photo unit. It also has keys to access all photo unit functions. Terminal response to photo unit commands simulate those of the typesetter.

Character widths are counted by the terminal exactly as they will be counted by the photo unit. Line-endings will be identical in both cases. If a command is entered that the photo unit could not perform, the terminal will reject the command and sound an audible alarm to alert the operator.

Photo unit functions begin and end with control symbols to distinguish them from data that is to be typeset. Some parameters are entered by individual keystrokes while others can be entered by specially programmed keys.

The terminal has two operating modes: Compose and Edit. In the Compose or counting mode, the width of each character entered is subtracted from the current line length. The values displayed in the message line represent the status of parameters referenced to the cursor in its present position. The cursor can be moved back and forth within a line without affecting the Compose mode. Vertical movement back through two previous line endings causes the Edit Mode to be invoked. Data entered in the Edit Mode updates the character count but character widths are temporarily ignored. Normal operation is restored by keying *Compose/Execute.*

In subsequent pages, descriptions of terminal responses are dependent on the Compose mode being in effect.



The four basic parameters are Line Length, Font, Point Size and Line Space. Four keys on the extreme right of the top row are dedicated to these functions.

At the start of each new job, default values have been assigned to the four main parameters as follows:

PS=8.0 LS=8.0 FT=1* LL=15.00

Other default values are as follows:

Letterspace	ON	3,1	Spaceband		12,8,30
Hyphenation	ON		Char.Comp.	OFF	
Pseudo Italic	OFF	12.0R	No Flash	OFF	
Multicolumn	OFF		Reverse Video	OFF	

Any parameter changes made by the operator will override the default values and will remain in effect for the remainder of the job, or until another change occurs. The status of any parameter not displayed in the message line can be examined by keying *File Function/Compose/Execute.* The listings that appear on the screen represent their current status *up to the location of the cursor* in the associated screen display. The parameter index can be cleared with the *Abort* key on the lower left of the Editing keypad.

*The first font listed in the font dress becomes the default font.

Display Parameter Index

PARAMETER DISPLAY

In the Compose mode, the status of all parameters *up to the location of the cursor* can be examined at any time by calling up the Parameter Index. This is accomplished by keying *File Func/Compose/Execute.* The first page of the menu lists the status of eighteen programs. Where applicable, O indicates ON; X indicates OFF. Values currently in effect are also displayed.

Display Parameters (PAGE 1)

$P_{\bar{S}}$ 10.0 $L_{\bar{S}}$ 11.0 $F_{\bar{T}}$ 83 $L_{\bar{L}}$ +5.00 A T_B XX J D=7.08 I 3122

Main Line Length:		15.00	Spaceband:		12,08,30
Letter Spacing:	O	03,01	No Flash:	X	
Set Size:		008.0	Pseudo Italic:	X	12.0R
Auto Compensation:	X	1	Hyphenation:	O	1
Character Comp:	X	00	Reverse Video:	X	
Kerning:	X		Multicolumn:	X	
V Quad Mode:	X		Column Depth:		000.00.0
VIS Count:		0000,0000	Galley Depth		000.00.0
Excess V Space:		000.00.0	SRP		0000,0000.0,00.0

To address the second page of the index, press Scroll Up. This page show the status of Indents and Cut Runarounds.

Display Parameters (PAGE 2)

$P_{\bar{S}}$ 10.0 $L_{\bar{S}}$ 11.0 $F_{\bar{T}}$ 83 $L_{\bar{L}}$ +5.00 A T_B XX J D=7.08 I 3122

Indent: X 00.00 Cuts In Use: NONE (or R1, L2, etc.)

Cut Runarounds:

CR01,022.00,006.00,05.00	CL01,043.00,006.00,05.00
CR02,067.00,006.00,02.06	CL02,067.00,006.00,02.06
CR03,000.00,006.00,00.00	CL03,000.00,006.00,00.00
CR04,000.00,006.00,00.00	CL04,000.00,006.00,00.00

Press Scroll Up again to access Page 3, which displays the status of the tab program. All 20 tabs are listed and those that have been defined prior to the position of the cursor will display parameters in each field. The letter "F" in the indent field identifies a floating tab.

Display Parameters (PAGE 3)

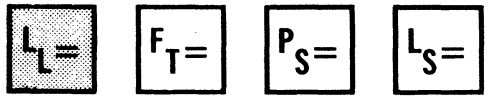
$P_{\bar{S}}$ 10.0 $L_{\bar{S}}$ 11.0 $F_{\bar{T}}$ 83 $L_{\bar{L}}$ +5.00 A T_B XX J D=7.08 I 3122

Tabs:

01,00.00,05.00L	02,F,05.00L	03,10.00,05.00L
04,13.00,06.00R	05,19.00,06.03R	06,27.06,03.06R
07,F,01.06J	08,00.00,00.00J	09,00.00,00.00J
10,00.00,00.00J	11,00.00,00.00J	12,00.00,00.00J
13,00.00,00.00J	14,00.00,00.00J	15,00.00,00.00J
16,00.00,00.00J	17,00.00,00.00J	18,00.00,00.00J
19,00.00,00.00J	20,00.00,00.00J	

Previous pages can be redisplayed by pressing Scroll Down. Each page can be accessed in turn by Scroll Up and Scroll Down. The *Abort* key restores the display of screen data to its previous status.

Line Length Parameters



LINE LENGTH

This value is entered in picas and points and defines the width of the current work area. As text is entered, the width of each character and wordspace is subtracted from the line length. The LL value displayed in the message line represents the current remainder. Default value is 15 picas.

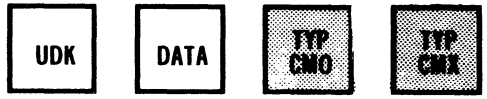
Line length entries must be made at the *beginning of a line* before any characters have been keyed for typesetting. They can be preceded by any other valid parameter entries.

Each programmed key has allowance for the maximum number of characters that can be entered in that field. The line length maximum is five: two positions for picas, one for a period, and two for points. When pressed, this key produces **<LL** on the screen and pauses for you to supply the necessary figures and is indicated by "Arg ?" (argument) in the message area. When all five positions are filled, the close Control symbol (>) appears, completing the function. The new line length will be displayed in the parameter line.

For example, to enter a line length of 30 picas, press the LL key/three zero period zero zero. This fills all five positions and produces the close Control symbol. A more efficient way to enter the same line length would be: LL key/three zero → (or LL key/three zero/Execute).

Pressing the right cursor (→) or *Execute* to indicate completion of a field applies to all programmed keys that include pauses for entry of variables. In functions which contain more than one field, → or *Execute* completes that field and produces a comma to separate fields. When the last field is complete, either key produces the close Control symbol (>).

If an error is made entering data for the Line Length (or any programmed key), deletions can be made with the Cancel Character key or a correction can be made by moving the cursor left (←) and overstriking the error. However, for UDK's that have several fields, the automatic features of the function are retained only if corrections are limited to the field currently being defined. The cursor will not move beyond the comma which precedes the current field. To correct an error in a previous field, data can be deleted with Cancel Character, or the function can be aborted. The cursor can then be moved to the left to correct the error. The function must be completed manually.



Any function can be entered manually by typing the entire command. The open and close Control symbols are accessed by the TYP/CMO (Typesetter Command ON) and TYP/CMX (Typesetter Command OFF) keys, located on the top row, third and fourth keys from the left. To enter a parameter, press the < key, type the function letters, figures, field separators and the > key. For example, this procedure would be necessary to program a UDK for a font change to access a pi character, then changing back to the original font. Programmed UDK's are disabled while defining another UDK.

FONT

The Font key is used to select a type face for the current line and will remain in effect until another font is selected. A font change can be made at any valid point in a line. For example, a font cannot be selected while another function is being defined or immediately after a floating accent.

The Font key is located on the top row to the right of the Line Length key and has provision for four digits. When pressed, the Font key produces **<FT**. A pause occurs for you to enter up to four digits. If fewer than four digits are entered, → or *Execute* produce the Control symbol > and complete the function. For example **<FT1>**, **<FT230>**, etc. The message line will display the font number selected.

If the font has not been loaded into the terminal memory, the message "FT?? Error" will be displayed. To display a list of fonts currently loaded in the terminal, press Display, type the letters "FT", then press *Execute*. The font identification number appears to the left of the font name on the screen. The index can be scrolled up and down to expose fonts not initially displayed. Press the *Abort* key to restore the previous screen display.

If a font is not specified at the start of a job, the default font will be in effect. The default font is the first font that was loaded into the current terminal memory and is displayed at the top of the font index.

Font changes can be made wherever necessary *within a block of copy.* This type of change is usually made within the same family to access the companion *italic* or **bold face** version. Font changes to different styles of type are usually made on a line-by-line basis.

Florentine Script abcde 12345

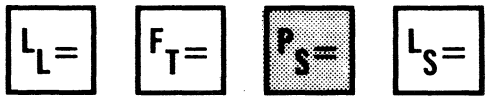
Univers Medium abcde 12345

Univers Bold Condensed abcde 12345

Souvenir Demi abcde 12345

Garth Graphic abcde 12345

Point Size Parameters



POINT SIZE

The Point Size key is on the top row to the right of the Font key and controls the height of characters which immediately follow. A point size may be selected at any valid place in a line and will remain in effect until another point size is selected. Point sizes for the 8400 are 5 to 72 point, graduated in half-point increments (5, 5½, 6, 6½, etc.) Standard 8200 point sizes are 5 to 36 point, also in half-point increments.

To select a point size, press the Point Size key. This produces **<PS** and pauses for you to enter one or two figures for whole points, followed by a period and one figure for half-point sizes. Completion of all values or pressing → or *Execute* completes the function. Examples: **<PS11>** or **<PS13.5>**.

Half-point sizes are selected by **.5**. If this field is vacant or contains the figure zero, a whole point size is indicated.

The message line will update and display the size selected. If an invalid size is keyed, an audible tone sounds and the message "PS?? Error" is displayed. For 8400 and 8600 photo units, entry of a point size deactivates any set size previously in effect and establishes a set size equal to the point size.

POINT SIZE EXAMPLES:

This is 5 point type
This is 6 point type
This is 7 point type
This is 8 point type
This is 8.5 point type
This is 9 point type
This is 9.5 point type
This is 10 point type
This is 11 point type
This is 12 point type
This is 14 point type
This is 18 point type
This is 24 point type
This is 30 point
Thirty-six point
48 point
54 point
60 point
72 point

Set Size Parameters

SET SIZE (8400 and 8600)

This feature provides a means of changing the width and appearance of a point size by expanding or condensing characters. The height of characters is not affected.

When a point size is selected with the PS key, a corresponding set size is established. For example, 12 point has a 12 set; 18 point, an 18 set; 72 point, a 72 set. To condense type, a set size smaller than the point size is assigned. To expand type, a set size greater than the point size is assigned. Either entry must be made *after* the point size has been selected.

A set size is entered by UDK/w at any valid point in a line. This produces <SS on the screen. Enter the set size by typing the appropriate figures and → or *Execute.* This results in **<SS12.5>** or **<SS5>**, for example. *The program will use the set size rather than the point size for counting character widths.* The set size is displayed on the right side of the second message line. An invalid set size causes the message "SS?? Error" to be displayed in the same area. Subsequent selection of a point size disables the previous set size. The point size is then used for width calculations.

The set size and point size ranges are identical; 5 to 72 point in half-point increments. Any set-size/point-size combination is valid except that 5 point cannot be condensed and 72 point cannot be expanded. (The set size/point size range for 8600 systems is from 4 to 127½ point.)

NORMAL POINT SIZE

<PS14> Fourteen point type ABC abc 123

EXPANDED

<SS15> Expanded 1 point ABC abc 123
<SS16> Expanded 2 points ABC abc 123
<SS18> Expanded 4 points ABC abc 123

CONDENSED

<SS13> Condensed one point ABC abc 123
<SS12> Condensed two points ABC abc 123
<SS 9> Condensed five points ABC abc 123

NORMAL POINT SIZE RESTORED

<PS14> Fourteen point type ABC abc 123

NOTE: Fixed spaces and word spaces are proportional to the set size. Each is larger as the set size is increased and decreases as the set size is reduced. In the above examples, the Em Space at the start of each line has different values.

Line Space Parameters

LINE SPACE

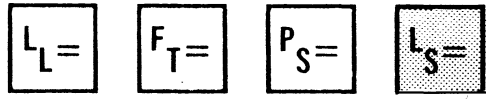
The Line Space key is on the extreme right of the top row. It controls the amount of paper that is advanced by the photo unit at the end of typeset lines. Its primary function is to make room for the line which follows. The value is specified in points and quarter-points. Default value is 8 points. The current value is displayed in the LS area of the message line and remains in effect until another line space is entered.

The depth count in the message line represents the accumulated depth in picas and points at the cursor location.

When data is being typeset, paper is advanced by the line space value each time the photo unit encounters an automatic or manual Return symbol. It is also the value activated by the Plus Line Space and Minus Line Space keys.

Press the Line Space key. This produces **<LS**. The system will now accept three figures for whole points, followed by a period and one figure for quarter-points. Fractional points are entered with **.1** for one-quarter, **.2** for one-half, and **.3** for three-quarter. If all five entries are made, the close Control symbol appears, completing the function. If fewer than five entries are made, pressing → or *Execute* signifies completion. The entry appears as: **<LS12>** for 12-point leading or **<LS12.3>** for 12¾-point leading.

Line space can be changed at any valid point in a line but generally is entered at the start of a line or just before a manual line-ending command. If two line space entries occur on a single line, the last entry takes precedence.



7-POINT TYPE, 7-POINT LEADING
If you recognize quality, if you value fine furniture, if you want to receive value for every dollar you spend, now is the time to shop at Home Furniture Company. We've included many of our most famous collection in this semi-annual clearance — beautiful furnishing of every style and period from the most distinguished names of the furniture world.

7-POINT TYPE, 8-POINT LEADING
If you recognize quality, if you value fine furniture, if you want to receive value for every dollar you spend, now is the time to shop at Home Furniture Company. We've included many of our most famous collection in this semi-annual clearance — beautiful furnishing of every style and period from the most distinguished names of the furniture world.

7-POINT TYPE, 9-POINT LEADING
If you recognize quality, if you value fine furniture, if you want to receive value for every dollar you spend, now is the time to shop at Home Furniture Company. We've included many of our most famous collection in this semi-annual clearance — beautiful furnishing of every style and period from the most distinguished names of the furniture world.

EIGHT-POINT LINE	LS12
TWELVE-POINT LINE	LS18
EIGHTEEN-POINT LINE	LS22
22-POINT LINE	LS18
EIGHTEEN-POINT LINE	LS24
24-POINT LINE	LS14
FOURTEEN-POINT LINE	

Line Ending Modes

LINE-END MODES

There are three line-end modes that can be selected by the operator: Automatic, Semi-Automatic and Manual.

Automatic Mode

This is the default mode and is in effect if no line-end mode is specified. In cases where you wish to select the automatic mode, press UDK and lower case "a" keys. This produces < and pauses for a single letter, in this case "a", to identify the automatic mode. This produces the remainder of the command: **<ALE>**.

The automatic mode is ideal for entering text material. The program automatically justifies lines and allows the operator to concentrate on producing the maximum amount of copy in the least amount of time. Once parameters have been entered, the operator enters text, inserts discretionary hyphens at strategic points and ends paragraphs. All other details are handled by the program. Lines are ended automatically on a word space or by hyphenating at a point that will produce the best possible character and word spacing. Copy justified in the automatic mode is completely flexible in the event any future editing or parameter changes are necessary. Simply enter parameters, edit and recompose.

The automatic mode will be used more frequently than either the semi-automatic or manual modes.

Semi-Automatic Mode

This mode is accessed by UDK/a/s, which produce **<SLE>**. It can be used during initial input of copy or after editing when a job is to be recomposed. Any line that can be justified on a word space, compound hyphen, DH or logic hyphen that occurs in the preferred justification range will be ended automatically by the program. In cases where a line cannot be ended on one of the above, overset will occur. The cursor will rest on the character which causes the overset.

If hyphenation is acceptable *immediately before the cursor,* press the word space key. If you prefer hyphenation EARLIER in the word, place the cursor on the character *to the right of* the preferred hyphenation point and press the word space key. After either selection, a soft DH/Return will appear where indicated and the remainder of the word will be carried over to the next line.

If neither choice is acceptable, move the cursor to the first letter of the current word and press the word space key. The space preceding that word will be converted to a soft Return and copy after it will be carried over. In all cases, the letter on which the cursor is placed will start the next line.

Manual Mode

The manual mode is accessed by UDK/a/m, which produce **<MLE>**. This mode should be reserved for jobs where the condition of every line is critical since it severely limits production. All the automatic features of the terminal relating to line-endings and hyphenation are disabled and become the responsibility of the operator for every line.

The manual mode can be used most effectively when a job is first keyboarded. Enter parameters normally, then enter text. As the end of a line nears, the operator must monitor the message line and try to end lines in the preferred range. If a word oversets and has to be hyphenated, the operator must furnish both the hyphen and the Return. If a line is to be ended between words, a Return must be entered where a word space would otherwise be keyed.

All line-endings entered in the manual mode produce hard Returns and will not be converted to word spaces if later editing is necessary.

Line Ending Modes

RETURN

This key is used to end a line and activate line space. It produces a full-intensity image on the screen, referred to as a hard return. Returns generated automatically by the terminal justification program are displayed at half-intensity and can be converted to word spaces if editing should cause them to occur in mid-line during later re-justification. Hyphen/returns provided by the hyphenation program are also displayed at half-intensity and may be discarded should they no longer occur at line-ending points.

Hard returns have permanent status and will never be altered during justification since they were placed there by the operator and over-ride the program.

Manual returns are often used with a quad key or an auto quad program and cause the line to be typeset accordingly. When no quad is involved, a return causes the line to be justified to the full measure.

When used to end a line in the tab mode, a return activates line space and positions the lens at the left margin of the *same tab column.*

When used to end a line which contains an Insert Character, it will fill the line with the character selected. If a line also contains one or more Insert Space commands, any remaining space will be inserted at those locations.

END LINE

End Line performs the same function as Return except there is no change in the depth count in the message line. On the photo unit, no leading occurs after the line is typeset and the lens returns to the left margin of the same line or column. End Line is used when no paper advance is to occur or where a leading amount less than the active line space is required.

Inserting cut-off rules between lines or blocks of copy is a typical application. In the example below, End Line is used to end the fourth tab column and is followed by two points of leading. Carriage Reset restores the overall line measure. Rules are ended with a Return, which activates normal line space. The Tab key can then be used to address Tab 1.

Year	Total	Local Aid	% state expend.
1973	2,804,820	794,778	28
1974	2,976,800	744,126	25
1975	3,629,068	1,010,564	28
1976	3,674,941	1,055,528	29
1977	4,122,340	1,069,427	26
1978	4,448,978	1,131,625	25
1979	5,139,640	1,438,153	28
1980	5,702,144	1,584,140	28
1981	6,062,329	1,613,114	27
1982	6,469,188	1,701,743	26

HYPHENATION PROGRAM

This program consists of a series of rules that are applied when an overset line occurs. The oversetting word is scanned for a hyphenation point that will assign the preferred value to word spaces in the line. If successful, hyphenation will occur at that point. The rest of the word will be carried over to the next line. If unsuccessful, the word is checked for a hyphenation point that will cause positive or negative letterspacing. If no such point is found, the word is carried over to the next line. The hyphenation program is ON at the start of each job. It can be turned OFF by UDK/H, which produces **<HYX>**. Hyphenation can be restored by UDK/h, which produces **<HYO>** .

ALTERNATE HYPHENATION PROGRAMS

During creation of User Program Disks in the System Generator, four hyphenation languages are optional. The base language (usually English) is active at the start of each job. An alternate language (French, German, etc.) can be activated by entering the number assigned in the System Generator in the following code sequence: **<HL2>**; **<HL3>** or **<HL4>**. Where necessary, **<HL1>** restores the base language.

DISCRETIONARY HYPHEN

The DH key is located on the bottom row to the right of the period. It provides a means of over-riding the hyphenation program for individual words, such as proper names or technical words that do not conform to logic rules.

DH's can be inserted where a hyphen would be acceptable and disables logic hyphenation for that word only. If the word does not overset the measure, DH's are ignored. A DH is recognized whether automatic hyphenation is ON or OFF.

DH's can be used negatively to prevent a word from hyphenating by placing one directly before or after a word.

JUSTIFICATION

The justification program uses several features to end lines when overset conditions occur. These features include hyphenation, positive and negative letterspacing and variable word spacing. Default values have been assigned to each but may be altered or deactivated by the operator to produce composition tailored to individual requirements.

The hyphenation program is on at the start of each job. If you turn hyphenation off, you eliminate one option the program has to justify lines efficiently. This can result in an increase in loosely spaced lines. A liberal use of discretionary hyphens is essential if hyphenation has been turned off.

Excessive word spacing can be the result if the maximum word space expansion is increased appreciably. A maximum of 54 units (minimum plus maximum expansion) will limit the word space to the same width as the point size and is generally accepted as a valid maximum word space.

If positive letterspacing is disabled and loose lines develop, the maximum allowable word space may be exceeded since this is the only remaining area the program has for the distribution of space in problem lines. If negative letterspacing is disabled, it removes one of the options the program has to justify marginally tight lines. With negative letterspacing on, the program can "steal" a little space from between letters and reduce the word space to its minimum value to fit an overset line into the measure. Without this capability, the entire oversetting word may have to be carried over and a loose line may result.

Reducing the preferred word space range can be a positive factor in obtaining well spaced lines but may be considered too tight for the last lines of paragraphs.

AUTOMATIC QUADDING

The key devoted to this function is located on the top row, to the right of Carriage Reset. When pressed, <A appears on the screen. A pause occurs for you to type a single letter to identify the style desired: **L** for left, **R** for right, **C** for center or **J** for justified. The close Control symbol then appears and completes the function: **<AL>**, **<AR>**, **<AC>** or **<AJ>**. The AJ mode is in effect at the start of each new job.

In the AJ mode, justification of lines is accomplished by expanding or contracting word spaces and by increasing or decreasing the space between letters. In Automatic Quad modes, all word spaces have the preferred value. Positive and negative letterspacing are deactivated. Excess space is distributed according to the program in effect.

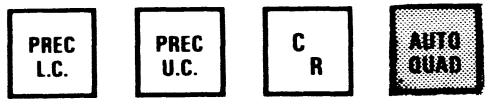
The Automatic Left mode produces copy with all lines flush to the left margin; blank space is placed to the right of lines.

The Automatic Right mode produces copy with all lines flush to the right margin; blank space is placed to the left.

The Automatic Center mode produces copy with lines centered between margins; blank space is divided evenly on either side of lines.

The Left, Right and Center Auto Quad modes can be used with hyphenation ON or OFF. Lines will be ended automatically by the program when overset occurs. On a one-line basis, the operator can override an auto quad mode with a manual line ending. For example, if Automatic Left is in effect, the program will accept Quad Right, Quad Center, Insert Space or Insert Character line endings.

The **<AJ>** symbol deactivates an auto quad mode and restore standard justification. It should be entered at the start of the line on which it is to become effective.



AUTOMATIC LEFT

Just a few winters ago, America was caught up in the throes of a natural gas crisis. Factories closed down on account of fuel shortages, and many observers were predicting the rapid-fire depletion of domestic gas reserves.

AUTOMATIC RIGHT

Just a few winters ago, America was caught up in the throes of a natural gas crisis. Factories closed down on account of fuel shortages, and many observers were predicting the rapid-fire depletion of domestic gas reserves.

AUTOMATIC CENTER

Just a few winters ago, America was caught up in the throes of a natural gas crisis. Factories closed down on account of fuel shortages, and many observers were predicting the rapid-fire depletion of domestic gas reserves.

AUTOMATIC QUADDING OFF

Just a few winters ago, America was caught up in the throes of a natural gas crisis. Factories closed down on account of fuel shortages, and many observers were predicting the rapid-fire depletion of domestic gas reserves.

MANUAL QUADDING

There are three keys on the main keyboard that are used to terminate lines and specify how blank space is to be distributed: *Left, Right* and *Center.*

When the *Left* key is followed by a line-end command, it causes the line to be typeset flush with the left margin. Blank space is placed on the right. It is used most frequently to end paragraphs. The message line will display the full line length or tab measure, depending on which command has been used to end the previous line.

Right is similar except that copy will be flush right and blank space will be placed on the left.

Center also is similar except that copy will be centered and blank space will be divided equally and placed on either side of copy in the line.

Any manual quad key can be used on a one-line basis to over-ride an auto quad or a tab quad of a different nature.

The word space in lines ended with a manual quad will be in the preferred range, space permitting. No letterspacing will occur in manually quadded lines.

Quad left copy */QL/Ret.*

Quad center copy */QC/Ret.*

Quad right copy */QR/Ret.*

The manual quad mode is especially useful when there is not enough consistency in line endings to warrant use of any of the automatic modes. In the style shown below, three different line-end conditions apply.

```
BOSTON                                  /QL/Ret.
      AAA MOTORS, INC.                  /QC/Ret.
                                        /Ret.
1130 Beacon St.          646-1730       /Ret.
BEMIS                                   /QL/Ret.
      XYZ MOTORS, INC.                  /QC/Ret.
                                        /Ret.
123 Main St.             842-2400       /Ret.
```

Spacebands (Interword Space)

SPACEBAND SET

This function establishes the following three standards that are used by the program during line-ending calculations:

1) *Minimum Wordspace* — This value is entered in units and represents the least number of units that will ever be placed between words for justification. (Units are relative to the point size in all three fields.)

2) *Preferred Wordspace Expansion* — This is also entered in units and represents the number that when added to the minimum equals the unit value that would be considered ideal for justifying every line. The preferred amount is used uniformly in the automatic quad left, right and center modes, and in manually quadded lines where space permits.

3) *Maximum Wordspace Expansion* — also expressed in units. When this value is added to the minimum word space (with letterspace program ON), it represents the greatest number of units that can appear between words during justification, and should be larger than 1 and 2 above combined. After this amount is assigned to each wordspace, any remaining space is divided equally between letters.

Current default values for the above three fields are **SB12,8,30**.

To change one or more fields, enter values via the UDK and "s" keys. This produces ⟨**SB** and a series of pauses. Each pause is for the entry of figures to satisfy each of the three fields. Commas appear automatically to separate fields and the close Control symbol appears after the third field has been completed: ⟨**SB10,6,32**⟩. This entry would assign a 10-unit minimum, a 6-unit preferred expansion and a 32-unit maximum expansion.

In the example cited, each wordspace is assigned a 16-unit value (10 plus 6) and the line length is reduced by that amount (in points) as a wordspace is entered initially. As the line length decreases, messages are displayed indicating the following ranges:

Positive Letterspacing: If the line should end at this point, it will be letterspaced and a wordspace of 16 units or more will result. (Letterspacing must be ON with an active positive value.)

Preferred Range: The wordspace will be in the 16-unit range. No letterspacing will occur.

Negative Letterspacing: The wordspace will be less than 16-units and negative letterspacing will occur. (Letterspacing must be ON with an active negative value.)

Overset: The line is overset and cannot be ended at the cursor location. The oversetting word will either be hyphenated and the overset portion carried over to the next line, or the entire word will be carried over.

Depending on which range is used to justify the line, the original wordspace of 16 units may be reduced to 10 units or may be increased to 42 units.

In straight text, where line-endings are determined by the justification programs, all three fields are equally important. In the larger point sizes, where line-endings will generally be controlled by the operator, maximum wordspace expansion will seldom be a factor. The sum of the first two fields represents the preferred wordspace, and should be less that the value used for text material.

In the three Auto Quad Modes (AC, AL, AR) wordspaces will always have the preferred value. Positive and negative letterspacing are disabled automatically and the message line will not display any reference to either.

Letterspace Parameters

LETTERSPACE SET

This function is used to change the values currently resident as the maximum allowable letterspace factors, both positive and negative. The default values are three units positive and one unit negative. The letterspace program is on at the start of each new job and must be turned off if not appropriate.

To change the default values, press UDK and the cap "L" keys. This causes **<LT** and two pauses to occur. The first is for the positive factor, the second for the negative factor. Type two figures for each field or one figure and → or *Execute.* The complete command will then be displayed: **<LT2,1>** or **<LT3,0>**, for example.

The values entered in these two fields are used as necessary by the program during justification and define the number of units the program can add between letters (positive) or subtract from letters (negative) to justify lines.

The positive value becomes a factor on loose lines when the maximum value has been calculated for each wordspace and space still remains in a line. It represents the greatest number of units that can be inserted between letters to justify the line.

The negative value becomes a factor on tight lines when the minimum value has been calculated for each wordspace and the line is still overset. It represents the number of units that can be subtracted between letters to make the line fit within the measure.

These are alternatives the program has in solving a mathematical problem and are not necessarily exercised each time one is calculated. A higher-priority alternative may be selected. The values are user-defined and provide full control of the typographic appearance of your composition on a job-to-job basis.

If changes are to be made to the defaults, the adjustment should be made in one-unit steps. Typeset enough copy on a fairly narrow measure to see if the results are acceptable.

LETTERSPACE ON/OFF

This function enables and disables the factors resident in the positive/negative letterspace program and defaults to the ON mode at the start of each job.

It can be turned OFF and ON by the UDK and lower case "ell" keys which produce **<LT** and a pause for one letter: X for OFF or O for ON.

In the ON mode, as data is keyboarded the program displays the justification status of the line at the cursor position: positive letterspacing; justification range; preferred range; negative letterspacing, and overset. The photo unit uses the mode selected by the program to typeset each line.

In the OFF mode, only the justification range, preferred range and overset messages are posted. When data is typeset in this mode, no letterspacing will occur, either positive or negative.

Fixed Space

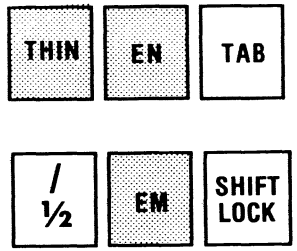
FIXED SPACES

Three fixed spaces can be accessed from the main keyboard: Em Space, En Space and Thin Space. Each fixed space represents a uniform value as opposed to the word space, which can expand or contract in order to justify a line.

The **Em Space** (54 units) provides blank space equal in width to the current point size (or set size). The symbol □ is displayed on the screen for the Em Space, which is most often used for paragraph indents and for the alignment of text and tabular matter. In most type faces, the Em Space has the same width value as the Em Dash and fractions.

The **En Space** (27 units) provides blank space approximately half the width of the Em Space. The symbol □ is displayed on the screen for the En Space. In most fonts, the En Space has the same width value as figures and is used in tabular matter to align columns. It can also be used for paragraph indents.

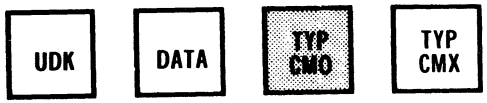
The **Thin Space** (18 units) provides blank space approximately one-third the width of the Em Space. The symbol □ is displayed on the screen for the Thin Space. In most fonts it has the same width value as the period, comma, colon, semicolon, open paren, close paren and the hyphen. It is used in tabular matter to align columns.



EM SPACE INDENT, 6 POINT
EM SPACE INDENT, 8 POINT
EM SPACE INDENT, 12 POINT
EM SPACE INDENT, 14 POINT
EM SPACE INDENT 18 PT.

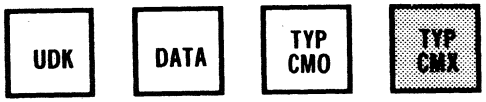
EN SPACE INDENT, 6 POINT
EN SPACE INDENT, 8 POINT
EN SPACE INDENT, 12 POINT
EN SPACE INDENT, 14 POINT
EN SPACE INDENT 18 PT.

THIN SPACE INDENT, 6 POINT
THIN SPACE INDENT, 8 POINT
THIN SPACE INDENT, 12 POINT
THIN SPACE INDENT, 14 POINT
THIN SPACE INDENT 18 PT.



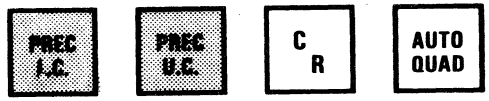
TYP CMO (Typesetter Command ON)

Located on the top row, third key from the left. This key is used as the opening symbol in a manually entered system command. It indicates that data which follows is program instruction rather than data to be typeset. Within a string of commands, it also indicates that the current command is complete and will be followed by another command.



TYP CMX (Typesetter Command OFF)

Located on the top row, fourth key from the left. This key is used at the end of a single manually entered system command and signals the program that the command has been completed. In a string of commands, it need only be used once, at the end of the last command.



PRECEDENCE L.C. and U.C.

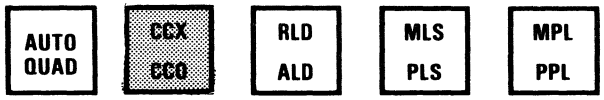
Located on the top row, fifth and sixth keys from the left. Both keys are used primarily with numeric keys to access pi characters. Pi characters vary from one layout to another, so the screen image that is produced by any two-key sequence will depend on which layout character set has been chosen for display.

As long as a layout has been loaded into the system font dress, the width of all characters on that layout will be counted correctly for justification and characters will be typeset correctly, but only those pi characters in the selected character set will be displayed correctly. For instance, if the 501 character set has been loaded and a font with the 500 arrangement is addressed, Precedence L.C. followed by a figure key will display the corresponding 501 character but will otherwise count the width of an inferior number and typeset it correctly as well.

Alpha-numerics for all layouts will be displayed correctly in all instances.

To access a Precedence character, refer to the appropriate keyboard layout to determine where it is located. Press the indicated Precedence key; this produces no image in the text area of the screen but causes PRECEDENCE LC (or UC) to appear in the message area; then press the numeric key. This causes the pi character image to appear in the text area on the screen.

When followed by a Return code, either Precedence key produces Carriage Reset, which deactivates the Tab program. However, a key has been pre-programmed to access this function with a single keystroke.



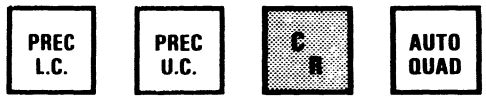
CCO/CCX (Character Compensation On/Off)

This key is located to the top row, to the right of Auto Quad. In the unshift position it produces <CCO> on the screen and turns character compensation ON. In the shift position, it produces <CCX> and turns character compensation OFF. In order for CCO to have any effect, a compensation factor must previously have been defined with <CCxx>, with xx's representing one-ninth points. This function is accessed via the UDK and the lower case "c", described below.

CHARACTER COMPENSATION SET (CCxx)

Character compensation is the uniform reduction of space between letters. It is generally used in large point sizes, 14 point and up. The amount of compensation is specified in one-ninth point increments by UDK and the lower case "c" key. This produces <CC and pauses for a two-digit entry. When supplied, the close Control symbol appears, completing the function. Example: <CC05>. Merely defining the factor does not implement it. It must be turned on by the <CCO> key. It can then be alternately turned off and on by <CCX> and <CCO>. If used in text sizes, the factor should be considerably smaller than that used for display sizes, 01 or 02, for instance. Once turned on, character compensation will remain in effect until <CCX> is encountered.

CC00 Character Compensation
CC02 Character Compensation
CC04 Character Compensation
CC06 Character Compensation
CC08 Character Compensation



CARRIAGE RESET (CR)

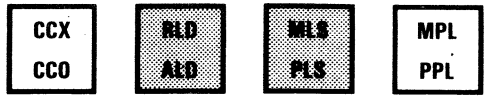
Located on the top row, to the right of the Precedence keys. When pressed, the CR logo appears on the screen. This deactivates the Tab program and restores the primary line length. This function can also be accessed by Precedence and Return. The latter sequence must be used to redefine the Carriage Reset UDK if it is to be stored in another location.

AUTOMATIC VARIABLE CHARACTER COMPENSATION

AVCC is an optional feature which provides three graduated character compensation programs entitled *Book Set, Ad Set* and *Tight Set.* It is defined by UDK/C, which produces <ACS and a pause for a single digit. The figure 1 identifies Book Set, 2 identifies Ad Set and 3 identifies Tight Set. The program must then be activated by <ACO>. It is deactivated by <ACX>. A detailed description appears in the AVCC portion of this manual.

NOTE: Codes to activate and deactivate AVCC (ACO and ACX) are accessed by the lower case "x" and upper case "X", respectively. If these codes are not included in default UDKs for your current program, they can be stored in a supplementary UDK file and merged with default UDK's.

Advance Leading



ALD/RLD (Advance/Reverse Lead)

Located on the top row, to the right of CCO/CCX, this key produces forward and reverse movement of photographic paper during typesetting. In the unshift position, paper will move forward. In the shift position, paper will reverse direction. Pressing the key produces **<ALD** or **<RLD** and pauses for up to six characters to identify the amount in points and quarter-points. When six characters have been entered, the close Control symbol will appear: **<ALD0034.0>**. If fewer than six characters have been supplied, press the right cursor key → or *Execute* to access the close Control symbol.

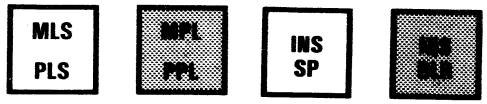
When data is typeset, paper moves at the point where the commands occur. Both forward and reverse commands can appear in the same line and can be repeated as often as necessary. The maximum amount of leading in either direction is dependent on the associated photo unit.

In the Compose mode, all leading commands update the depth count in the parameter line.

PLS/MLS (Plus/Minus Line Space)

Located on the top row, to the right of ALD/RLD. This key is used to advance photographic paper forward or reverse by the current line space value. In the unshift position, the symbols **<PLS>** appear and paper will move forward. In the shift position, the symbols **<MLS>** appear and paper will move in the reverse direction.

When data is typeset, paper moves at the point where the commands occur. Both commands can be used in the same line and can be repeated as often as necessary.



PPL/MPL (Plus/Minus Point Lead)

Located on the top row, to the right of PLS/MLS, this key is used to move photographic paper forward or reverse in one-point increments. In the unshift position, each depression produces **⟨PPL⟩** and paper will move forward one point. In the shift position **⟨MPL⟩** appears and paper will reverse one point.

When data is typeset, paper moves at the point where the commands occur. Both commands can be repeated in the same line.

NOTE: The limitation of Reverse Leading will vary, depending on the phototypesetter. The MCS8400 is limited to 14 inches (84 picas/1008 points). Although the MCS8400 will reverse lead up to 1008 points, reverse leading beyond 72 points is not recommended. To ensure reverse leading accuracy, the Multiple Column leading option (MCO) should be utilized.

INS BLR (Insert Baseline Rule)

Located on the top row, to the right of Insert Space, this key produces the symbols $\overset{I}{C}\ \overset{B}{L}$. The photo unit will fill the remainder of the current line with connecting baseline rules. The rules will be typeset in the current point size. The weight of rules is proportional to the point size: light rules in small sizes, heavier rules as the size increases.

In cases where the inserted rules do not fill the line, the fractional remainder will be distributed as blank space in the following priority order:

1) At Insert Space locations, if any;
2) Between words, if no Insert Space occurs;
3) On either side of the rules, if no Insert Space or word spaces occur, or between characters if no word space is present.

TAB PROGRAMS

Twenty tab positions with variable indents, measures and quad conditions can be stored in the terminal at any given time. Each tab postion is identified by number, 01 to 20. The combined measures of all adjacent tab columns must not exceed 70 picas. There are three tab forms; sequential tabs, auto tabs and floating tabs. All three can be used in composing a job but must use the same series of 20 numbers.

Sequential tab columns are defined by use of UDK and the lower case "t". When the two-key combination is entered, it produces **<TB** and stops for the operator to enter one or two figures to identify the tab number. A comma will be added by the program: **<TB01,**

The operator now enters the tab indent, which consists of two figures for picas, a period, and two figures for points. Another comma is added: **<TB01,10.00,**

The next field identifies the tab measure and is structured the same as the indent. It will generate another comma when completed: **<TB01,10.00,13.09,**

Type the quad letter (L for left, R for right, C for centered or J for justified) and the close Control appears to complete the tab definition: **<TB01,10.00,13.09,C>**

In the above example, each field has the maximum number of entries. In cases where zeros represent no value, they can be omitted. The cursor right is then used to complete the field: **<TB1→,10→,13.9→,C>**

After the close Control appears, if a structural error was made when defining the tab, an audible tone will sound and a message will be displayed in the parameter line. The cursor will rest on the close Control symbol. Examine the tab entries and when the error is found, move the cursor to the left and make the correction.

When all tabs have been defined, call up the parameter index and check the tab entries. The Abort key restores the screen to its previous status.

FLOATING TABS

A floating tab is one for which no indent value will be specified. It has a tab number, measure and quad letter only. Eliminating the indent allows you to use the same tab number more than once at different locations. Floating tabs are used when a tab job requires more than 20 columns. If fewer than 20 columns are required, the sequential tab format should be used.

To define a floating tab, press UDK and the upper case "T". When the two keys have been entered, **<TF** appears and a pause occurs for the tab number. A floating tab should be assigned the number of the column in which it first occurs. When the tab number has been entered, the program furnishes a comma and waits for the measure. Enter the measure and the field separator comma will appear. Enter the quad letter, which will generate the close control symbol. Example: Floating tab 7 with a 4 pica measure and centered copy: **<TF7,4,C>**.

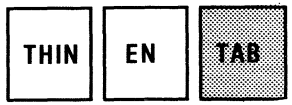
When addressed *sequentially* with the Tab key, Tab 7 will be indented by the combined indent and width of Tab 6. When required for a subsequent column, it must be addressed by its tab number **<CT7>**. It will then be indented by the combined indent and width of the tab column which preceded it. The tab measure and quad status will be the same as in the first instance.

If a *blank tab column* is required, Insert or Fixed Space (Em, En, Thin) must be entered as text to complete the floating tab.

Tabbing Functions

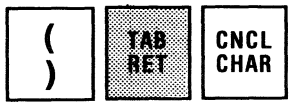
CALL TAB

Once tabs have been defined, they can be activated by the Call Tab key, located to the right of Cancel Word. This produces <CT and pauses for the two-digit address, or one digit and → or *Execute.* When supplied, the close Control symbol appears <CT1> and the tab program is activated. Call Tab and the tab number can be used to address a tab column out of sequence. For example, if you are in Tab 1 and copy next occurs in Tab 9, <CT9> will skip the intervening columns and address Tab 9. Call Tab also activates justification of the current tab column.



TAB KEY

The Tab key, to the right of the En Space, also activates the Tab program and addresses Tab 1. The tab number, line length and left margin are displayed in the second message line. Once tabs are activated, the Tab key is used to end the current column and address the next column in sequence. If no copy is to occur in a column, and if Floating Tabs are not involved, you can step through a blank column by pressing the Tab key again.



TAB RETURN

When copy for the last column on the right has been entered, use Tab Return to end that column. This also activates line space and moves the lens to the left margin of the *overall measure.* Use the Tab key to address Tab 1, or the Call Tab key and the tab number to address another column.

RETURN KEY IN TABS

When consecutive lines are to be entered in the same tab column, lines are ended with a Return, either a manual Return entered by the operator or an automatic Return furnished by the program. The Return ends the line, advances line space and moves the lens to the left margin of the *current column.* The final line-ending for the column will depend on where the next column is located.

If the next sequential column is to be addressed, the Tab key is used. This may also require reverse leading to access the proper vertical position. When the last column has been completed, Tab Return ends the line, advances line space and moves the lens to the left margin of the overall measure.

REGIONAL OFFICES

ATLANTIC
1901 North Moore St.
Suite 506
Rosslyn, Virginia 22209

PACIFIC
16421 Carmenita Road
Building 5
West Business Park
Cerritos, CA 90701

SOUTHERN
2527 Lantrac Court
Decatur, Georgia 30035

NEW YORK METRO
Room 1888
2 Pennsylvania Plaza
New York, N.Y. 10001

CENTRAL
Suite 57
415 West Golf Road
Arlington Hts., Ill. 60005

CANADA
CG Canada, Limited
1122 Aerowood Drive
Mississauga, Ontario
Canada L4W2K7

DEACTIVATING TABS

To deactivate tabs and restore the primary line length, depress the Carriage Reset UDK (or Precedence, Return). If parameter continuity is maintained from page to page, defined tabs can be activated on any page in the same job.

AUTOMATIC TABBING

In conjunction with the standard tab operation, the MCS can automatically establish tab positions. The Auto Tab is positioned in a text line after the calculation for justification or quadding. Since the tab is established after calculation, it can be inserted anywhere, and will automatically enter tab parameters (Indent and Tab Measure). Auto Tabs are used to define tab positions within text lines wherein the tab values cannot be readily pre-defined. For example, a tab within a justified line, variable indents, underscores, etc.

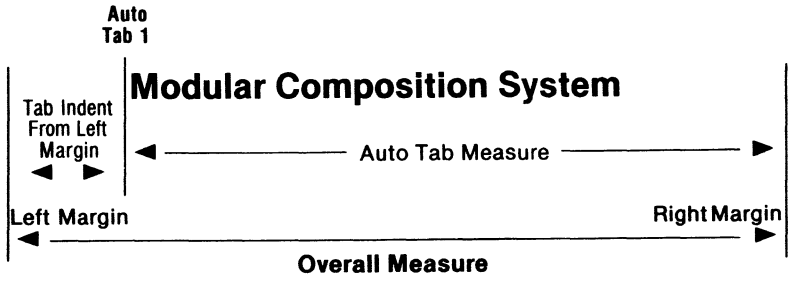
All functions and limitations of the standard tab are applicable to Auto Tabs. Either standard or Auto Tabs can be inserted within a text line, limited to 20, the maximum number of tabs. The Auto Tab lines, like the standard tab, can be designated to automatically quad or justify the tab lines. The tabs are identified by Auto Tab number (1 - 20) and accessed with the *Call Tab* command.

Auto Tab Line Measures

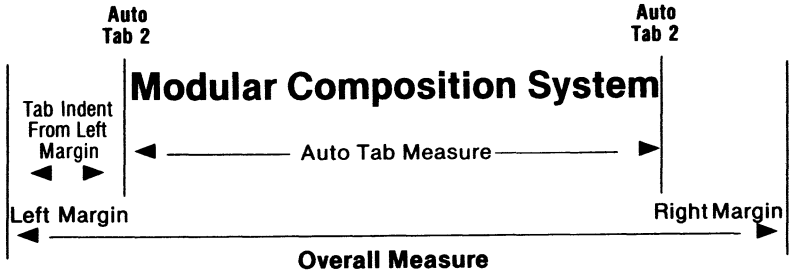
An Auto Tab, if placed at the beginning of text, will, after calculation, establish a tab measure *equal to the total line measure, minus the amount of tab indent from the left margin* (See example A).

Defining an Auto Tab with the *same tab number* on both sides of the manuscript copy will establish a tab measure *equal to the width of the copy* (See example B). The Auto Tab inserted in the beginning of the line is established at the following character; the Auto Tab inserted at the end of the line sets the tab after the character immediately preceding the duplicated Auto Tab number.

Example: A Auto Tab inserted at the beginning of a centered line.



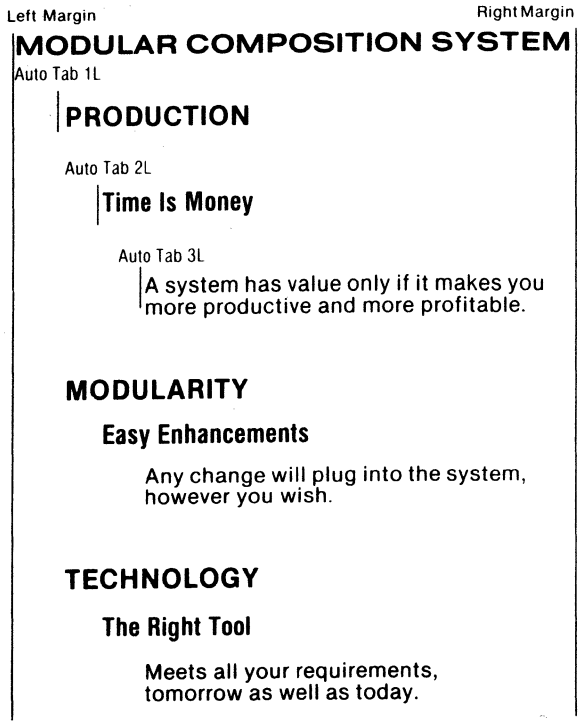
Example: B Auto Tab equal to the measure of text on centered line.



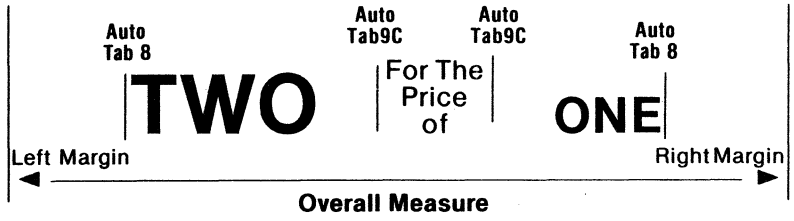
Auto Tab Quadding

The definition of an Auto Tab can include a quadding command to position manuscript copy within the tab. The Auto Tab commands are **(Auto Tab #,L) Quad Left; (Auto Tab #,C) Quad Center; (Auto Tab #,R) Quad Right.** The letter, immediately following the tab number, positions the text.* *If no command is inserted, the copy will be justified.* The defined quadding can be over-ridden where required on a one-line basis.

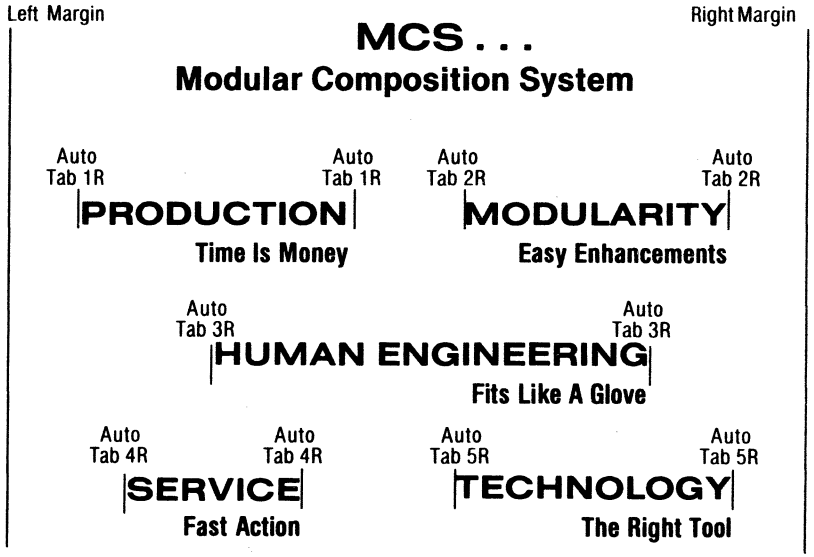
Example: Auto Tabs Defined Quad Left


Left Margin
Right Margin
MODULAR COMPOSITION SYSTEM
Auto Tab 1L
PRODUCTION
Auto Tab 2L
Time Is Money
Auto Tab 3L
A system has value only if it makes you more productive and more profitable.
MODULARITY
Easy Enhancements
Any change will plug into the system, however you wish.
TECHNOLOGY
The Right Tool
Meets all your requirements, tomorrow as well as today.

Example: Multiple Auto Tabs, Justified and Centered



Example: Auto Tabs Defined Quad Right



*NOTE: The quad code(s) need only be entered once and in either the left or right position when defining Auto Tabs.

Insert Space

INSERT SPACE

Located on the top row, to the left of Insert Baseline Rule, this key produces the symbols **<IS>** when depressed. *Insert Space* performs two basic functions. Its primary use is to control the distribution of relatively large amounts of blank space in typeset lines such as the following:

Left**<IS>** Right

Left**<IS>** Center**<IS>** Right

One**<IS>** Two**<IS>** Three**<IS>** Four

A second application occurs in lines which contain one or more Insert Character commands but have no wordspaces. *Insert Space* is used to control the distribution of fractional amounts of space that remain when the deficit in a line is not evenly divisible by the width of the inserted character. It is also recommended in Insert Character lines even when wordspaces occur to provide uniform space between words and eliminate possible letterspacing.

Name <IS>

Address <IS>

City <IS> State

In lines with no wordspaces, this feature enables the right-hand alignment of the inserted characters that might otherwise be ragged.

December <IS> 123,456

January <IS> 234,567

INSERT BLANK SPACE

There are instances where it is convenient to be able to insert a proportional amount of blank space in a line that also contains an Insert Character command. The Insert Space key does not perform that function. In lines with an Insert Character it controls where the *fractional deficit* is to be placed. Blank space can be obtained by a series of fixed spaces or by use of Tab columns but a much simpler method is available.

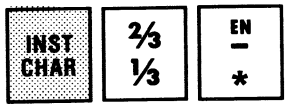
The program will insert a proportional amount of blank space in lines with an Insert Character by keying Insert Character, Em Space* at the point where the blank space is to appear. Where equal amounts of blank space and baseline rule are to appear in a line, the deficit at the end of the line is divided in half. One-half of the line will be blank space and the other half will be baseline rules. If two rule insertions and one blank space insertion are keyed, one-third of the deficit will be assigned to each insertion. Examples:

Insert Character/Em Space; Insert Rule; Return

Insert Rule; Insert Character/Em Space; Insert Rule; Return

*The program will also recognize *Insert Character/En Space* or *Insert Character/Thin Space* for the insertion of blank space.

Insert Characters



INSERT CHARACTER

This key is located at the extreme left on the bottom row. It is used to typeset a continuous series of the selected character to fill the remainder of the line measure. If more than one Insert Character command occurs in a line, a proportional amount of space will be allotted to each.

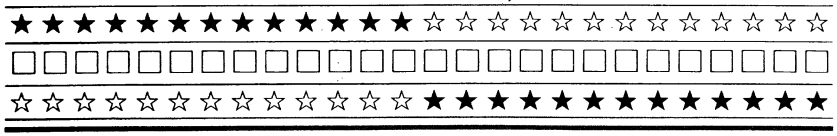
Insert Character is keyed, followed by the character that is to be inserted. The insert may be incorporated with text or it may be used independently. Typical insertable characters include the Period, En leader and Baseline Rule. However, virtually any character can be used, with the exception of accents, the fraction bar or any other character that has a zero width value.

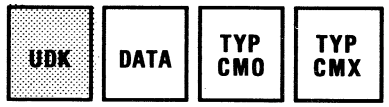
If the inserted characters do not completely fill the line, the fractional remainder will be distributed as follows:

If one or more Insert Spaces occur in the line, the deficit is divided equally between them, or the entire deficit is assigned to a lone Insert Space.

If no Insert Spaces occur, the deficit is distributed equally between word spaces.

If neither Insert Spaces nor word spaces occur, the deficit is divided equally on either side of the insert(s).





UDK

The letters UDK represent User Defined Key, which is located on the left side of the top row. This key is used in combination with 18 alpha keys to access system commands with a minimum of keystrokes. In some cases a complete function is accessed by the two-key combination while in others a pause occurs for the operator to supply variables. Where indicated, shift and unshift keys access different though related functions.

When entering commands, you can bracket each command between Control symbols or you can string commands.

A command string consists of a series of commands with open Control symbols only, except for the last command which must have both the open and close Control symbols. Instead of ending a command with the right arrow, select the next command. This completes the first command and addresses the second but no close Control symbol appears. Repeat this procedure until the last command in the string is complete. The right cursor or *Execute* completes the string. (See User Define Keys, Page 86).

Individual commands:

<SM22><PS12.5><LS14><FT18><HYX>

A command string:

<SM22<PS12.5<FT18<HYX>

Indent Parameters

INDENT SET

There are four indent styles available: Hanging Indent, Left Margin, Right Margin and Both Margins. Only one of the four styles can be defined and activated at any given time.

The style and amount of the indent is defined by UDK and the cap "I" keys. This causes **<I** to appear on the screen. A pause occurs for the operator to provide one letter for the style (H L R or B), two digits for picas, a period and two digits for points. The → or *Execute* may be pressed if fewer than five entries are made for the amount. This produces the close Control symbol and completes the indent definition: **<IL2.6>** left indent of 2 picas, 6 points.

Defining the indent does NOT turn it on. The indent will remain dormant until activated by UDK and the lower case "i" keys. This produces **<I** and a pause for the operator to type "O" for On or "X" for OFF. In this case "O" will be selected. This command must be entered at the beginning of the line on which the indent is to start. The Indent Off command should be entered at the start of the first line that will not be indented. *The line preceding either entry must be ended with a hard return.*

Hanging Indent

This style indent causes the first line of each paragraph to be set to the full measure. Runover lines will be indented on the left margin by the amount specified. The quad left at the end of each paragraph recycles the indent, causing each succeeding paragraph to be set in the same manner. The indent can be turned on and off by **⟨I0⟩** and **⟨IX⟩**.

The Hanging Indent is the only style that recycles. The remaining three styles treat each line exactly the same with respect to the indent in effect.

All four indent styles can be used with the main measure or with tabular columns already indented from the left margin.

The indent status can be examined by File Func/Compose/Execute, which causes the parameter index to be displayed on the screen. The style, amount and On/Off status of indents will be listed. ABORT restores the previous display.

⟨IH1.6⟩⟨I0⟩

BASEBALL — Boys 8 to 18. Play equivalent of three baseball seasons in one summer. Visits by baseball notables. Daily ball games plus batting cages and training areas.

UMPIRE SCHOOL — Boys 15 to 22. Learn the professional way. Drills, tests, umpire under game conditions.

TENNIS PROGRAMS — Boys 8 to 18. New 5-court indoor tennis complex, 4 outdoor courts. Teaching pros. Daily instructions. Ball machine drills, competition.

⟨IX⟩

The training programs listed above have a limited number of openings. Applications should be made early to ensure confirmed reservations.

Indent Parameters

Indent Left

This style causes all lines to be indented by the specified amount from the LEFT margin. Press UDK and the cap "I" keys. Type "L", the pica value, a period and the point value: ⟨**IL3.6**⟩, ⟨**IL3**⟩, etc.

To enable the indent, press UDK "i" and "O" ⟨**IO**⟩ at the start of the first line to be indented. To deactivate, enter ⟨**IX**⟩ at the start of the line where the indent is to stop. (If the line preceding either command has a soft return, convert it to a hard return.)

Indent Right

This style causes all lines to be indented by the specified amount from the RIGHT margin. Press UDK and the cap "I" keys. Type "R", the pica value, a period and the point value: ⟨**IR3**⟩.

To enable the indent, press UDK "i" and "O" ⟨**IO**⟩ at the start of the first line to be indented. To deactivate, enter ⟨**IX**⟩ at the start of the line where the indent is to stop. (If the line preceding either command has a soft return, convert it to a hard return.)

Indent Both

This style causes all lines to be indented by the specified amount from BOTH margins. In this instance, the line length will be reduced by double that amount. Press UDK and the cap "I" keys. Type "B", the pica value, a period and the point value: ⟨**IB1.6**⟩.

To enable the indent, press UDK "i" and "O" ⟨**IO**⟩ at the start of the first line to be indented. To deactivate, enter ⟨**IX**⟩ at the start of the line where the indent is to stop. (If the line preceding either command has a soft return, convert it to a hard return.)

LEFT MARGIN INDENT

The left margin indent program provides blank space at the left margin.

> This feature can be used to provide room for an illustration or for an initial letter. The depth counter in the message line can be used to calculate the depth of the

indent, which can be turned off when a sufficient amount of space has been generated.

RIGHT MARGIN INDENT

The right margin indent program provides blank space at the right margin. This feature can also be used to provide room for an illustration or one or more larger size characters. In effect, it shortens the measure by the amount of the indent. It can be reactivated at any subsequent point in the same job.

BOTH MARGIN INDENT

The both margin indent is often used to set a paragraph in bold face type.

> **The indent and change in type face relieves the monotony in a long column of type and help to retain the reader's attention.**

Indented paragraphs are often set in italics to indicate that the material is an extract from other data or to provide emphasis.

CUT LEFT/CUT RIGHT SET

There are eight cut programs that can be defined and used at the same time. Four are indented from the left margin and four are indented from the right margin. Each cut is referenced to the start of the job at the point where the depth reading equals zero.

Left cuts are defined by UDK and the lower case "o" keys, which produce the following:

⟨CL	Pause for figures 1, 2, 3 or 4 (address)
comma	Pause for depth at which cut is to start. Field consists of three positions for picas, a period and two for points (170.07 is maximum depth for this field).
comma	Pause for depth of the cut; same structure as previous field.
comma	Pause for width of cut (indent in picas and points).
⟩	Definition complete.

The procedure can then be repeated and three more left cuts and four right cuts can be defined. Right cuts are addressed by UDK and cap "O" and produce **⟨CR** rather than CL. Otherwise procedures are identical.

Although each cut is referenced to the start of the job, the cut definitions will usually be entered after text has been keyboarded. A location can be selected that appears to be acceptable. In the Compose mode, moving the cursor down through the copy updates the depth count so that a pica and point reading can be obtained as the starting point of a cut.

Press Limit, Scroll Down. Move the cursor just beyond the job parameters and press Compose, Execute and Insert. Press UDK and "o". Enter the address (1 to 4). Next enter the pica/point values for the starting point of the cut, its depth and the amount of indent.

If more than one cut is to be used, follow the same procedure for locating and defining the remaining cuts. Once defined, cuts turn themselves on at the depth indicated and off when the cut depth has been met. In each case, if the depth figure is not matched exactly, the next higher reading serves the same purpose.

Finally, after composing the entire job, examine the cuts to see if any bad breaks occur. A slight adjustment up or down in the starting point of a cut can be made. Recompose copy and examine cut positions again.

In the example below, the following codes were entered for the left and right cuts:

⟨CL1,4,4.6,6⟩ **⟨CR1,11.8,4,5⟩**

The wafer can be selectively masked against the ions either by a patterned oxide layer, as in conventional diffusion, or by a photoresist pattern. For example, phosphorus ions accelerated through a potential of 100,000 volts will penetrate the photoresist to a depth of less than half a micrometer. Wherever they strike bare silicon they penetrate to an average depth of a tenth of a micrometer. Thus even a one-micrometer layer of photoresist can serve as a mask for the selective implantation of phosphorus. As the accelerated ions plow their way into the silicon crystal they cause considerable damage to the crystal lattice. It is possible to heal most of the damage, however, by annealing the crystal at a moderate temperature. Little diffusion takes place at the annealing temperature, so that the ion-implantation conditions can be chosen to obtain the desired distribution. For example, a very shallow, high concentration of dopant can be conveniently achieved by ion implantation. A more significant feature of the technique is the possibility of accurately controlling the concentration of the dopant. The ions bombarding the crystal each carry a charge, and by measuring the total charge that accumulates the number of impurities can be precisely determined. Hence ion implantation is used whenever the doping level must be very accurately controlled. Often ion implantation simply replaces the predeposit from a high-temperature vapor. For example, the current exploration of the use of arsenic as a shallow n-type dopant in MOS devices coincides with the availability of suitable ion-implantation equipment.

NO FLASH ON (Non-Photo Mode)

The function **No Flash On** prevents the photo unit from flashing characters. While in the mode, the photo unit will escape the width of each character, providing the space, but characters will not be photographed. The function code is accessed with the User Define Key lower case "n." The letters NFO are displayed in the message line. No other terminal procedures are affected.

No Flash On is used to position data in the current line and allow space for other data that will be typeset in a subsequent line.

NO FLASH OFF (Photo Mode)

Once No Flash On has served its purpose, the No Flash Off (photo) mode can be activated in two ways:

1) Manually by UDK and the upper case "N" keys, which produce **⟨NFX⟩**. This method *must be used* to deactivate the non-photo mode in mid-line, where characters beyond that point must be typeset.

2) Automatically by any line-ending command (return, tab, tab return, end line). This situation occurs when the non-photo mode is in effect at the end of a line. This prevents inadvertently leaving the photo unit in a non-productive condition.

COMPUGRAPHIC LOW COST HIGH QUALITY PHOTOTYPE

⟨PS24⟨FT3⟨LS7⟨LL26⟩
⟨NFO⟩⟨PS24⟩COMPUGRAPHIC**⟨NFX⟩⟨PS7⟩**EnSpace LOW COST QL Ret.
⟨NFO⟩⟨PS24⟩COMPUGRAPHIC**⟨NFX⟩⟨PS7⟩**EnSpace HIGH QUALITY QL Ret.
⟨PS24⟩COMPUGRAPHIC**⟨PS7⟩**EnSpace PHOTOTYPE QL Ret.

DISPLAY ABC WIDEST LINE

⟨PS28⟨FT21⟨LL26⟨LS11⟩⟨NFO⟩DISPLAY**⟨PS10⟩**EM,ABC,QC,Ret
⟨PS28⟩DISPLAY**⟨PS10⟩**EM,WIDEST LINE,QC,Ret.

EXAMPLE WIDEST LINE XYZ

⟨PS28⟨F121⟨LL26⟨LS10⟩⟨NFO⟩EXAMPLE**⟨PS10⟩⟨NFX⟩**EM,WIDEST LINE,QC,Ret.
⟨PS28⟩⟨NFO⟩EXAMPLE**⟨PS10⟩⟨NFX⟩**EM,XYZ,QC,EndLine.
⟨PS28⟩EXAMPLE**⟨NFO⟩⟨PS10⟩**EM,WIDEST LINE,QC,Ret.

Flash Only

FLASH ONLY

This function assigns a width of zero to the character *immediately following the command.* It is accessed by the UDK and lower case "f" keys, which produce **<FO>** on the screen. It can be used to assign a width to a character different from the one normally assigned, usually a smaller amount.

When the photo unit encounters a Flash Only command, it photographs the next character but does not allow space for the character. If another character follows, it will be photographed in the same position. If followed by fixed space and a character, the carriage will move the width of the fixed space before photographing the character.

Flash Only can be combined with Space Only On (non-photo) to typeset one character and move the carriage the width of another character.

FLASH ONLY EXAMPLES

Market Ads **49¢ LB**

⟨PS48⟨FT12⟨LL26⟨LS24⟩49⟨FO⟩¢⟨PS12⟩LB,QC,RET.

Indent a period under quotes."
Indent a period under an asterisk.*
In each case, enter **⟨FO⟩** before typing the period.

Flash Only can be used to align figures at the right with reference symbols extending beyond the main measure or tab column.

Sample A 234
Sample B 276*
Sample C . . . . . 198□
Sample D 144

a . . . . . 1234* b . . . . . 2345 c 3456
1 . . . . . 1234 2 . . . . . 2345x 3 3456
A . . . . . 1234 B . . . . . 2345 C . . . . . 3456★

Enter **⟨FO⟩** before the reference character (asterisk, open box, etc.) Each will be typeset beyond the line measure.

Flash Only used to maintain centering position and disregard punctuation.

Plus F.E.T. Tax	Plus F.E.T. Tax

To maintain an optical center for alpha characters F.E.T. enter **⟨FO⟩** before the last period, as shown in the first column.

FORWARD POINT

This function causes horizontal blank space to be added to a line. Space is added where the function codes occur and is generally used to increase the space between letters. It can also be used to reduce the LL= reading to a particular value. It is accessed by UDK and the lower case "p", which produce **<FP** on the screen. A pause occurs for two figures (or one figure and → or *Execute*) to identify the number of points that are to be added. A point is a fixed amount and has the same value in all point or set sizes.

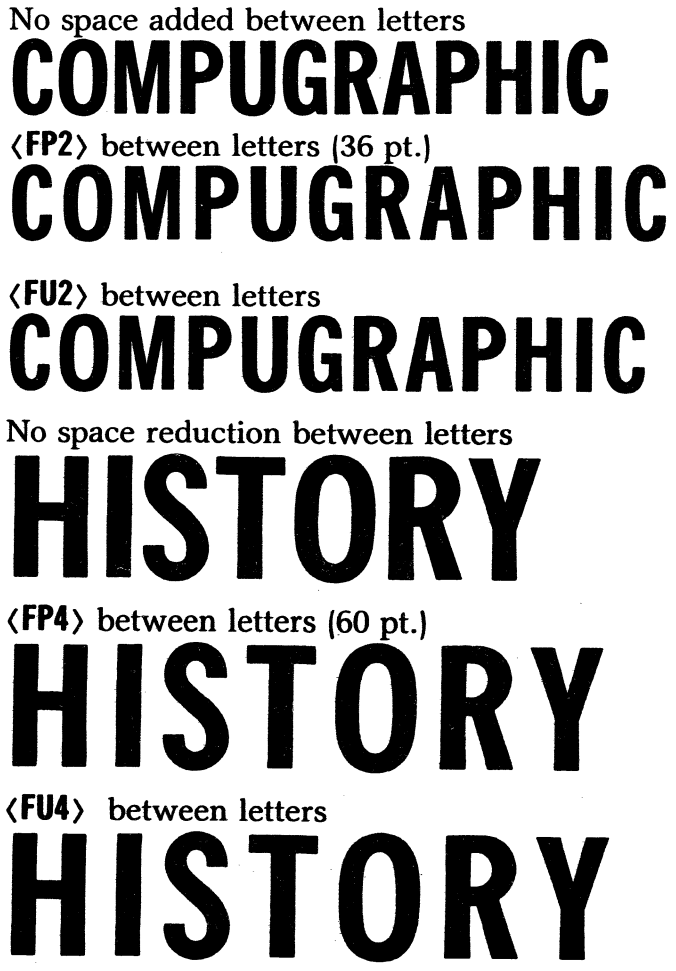
FORWARD UNIT

This is similar to Forward Point except that it is incremented in relative units rather than in points. It is accessed by UDK and the lower case "u", which produce **<FU** on the screen. Two figures (or one figure and → or *Execute*) identify the number of units that are to be added. Each unit has a width 1/54th of the point size or set size.

WHICH TO USE

One aid in comparing the point and the unit is to use 54-point as the standard. The Em Space for this size is 54 points wide; therefore, a unit in this size is one point wide.

The unit is less than a point in sizes below 54-point and greater than a point in sizes above 54-point. The unit is equal to one-half point in 27-point and one-quarter point in 13.5-point.



Manual Letterspacing

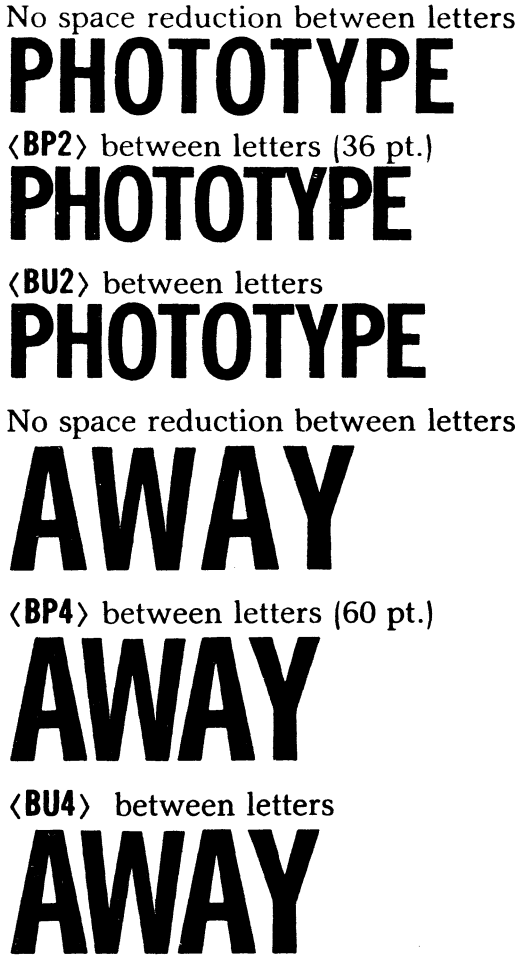
BACK POINT

The purpose of this function is to subtract white space from a typeset line at the point of insertion. It is accessed by UDK and the upper case "P" keys which produce **<BP** on the screen. A pause occurs for two figures (or one figure and → or *Execute*) to identify the number of points that are to be subtracted: **<BP2>**. A point is a fixed value and has the same width in all point sizes.

This function is commonly used between letter combinations which might appear to be letterspaced. It cannot be used at the *beginning* of a typeset line.

BACK UNIT

This serves the same purpose as Back Point except that units are specified rather than points. A unit is 1/54th the width of the point size in use and is a relative value. Back Unit is accessed by UDK and the upper case "U" keys, which produce **<BU**. A pause occurs for two figures (or one figure and → or *Execute*) to identify the number of units that are to be subtracted: **<BU1> <BU3>**. This is an invalid command at the *beginning* of a line.



Pseudo Type Style

PSEUDO ITALIC (8400 Only)

This function defines the degree to which roman type faces are to be electronically italicized. A default value of 12 degrees of right slant is registered at the start of each new job but is not active. The program will accept degree changes in a range of from 1 to 31 degrees.

To address this function, UDK and the cap "V" keys are pressed and produce **<PI**. Enter two digits (or one digit and → or Execute) to define the degree of slant, after which the close Control symbol appears: **<PI13>**, **<PI9>**.

Pseudo italic should be applied only to the alphanumerics and punctuation of roman type faces. Pi characters (bullets, stars, math symbols, etc.) should be set in the roman style. Pseudo italic should not be applied to italic or script type styles or ornamental characters.

Caution must be taken in applying pseudo italic to extra-wide type styles in the upper range of point sizes. This can result in clipping the right edge of characters that exceed the width of the window through which characters are photographed. (Characters that would photograph normally in the roman style.)

The program requires the Pseudo Italic On command **<PIO>** to activate it and the Pseudo Italic Off command **<PIX>** to deactivate it.

Characters can also be slanted to the left from 1 to 31 degrees by entering the letter "L" after the figures **<PI20L>**. To return to normal pseudo italic, an "R" must be used for the next occurrence **<PI12R>**. The status of this function can be examined in the parameter index (File Function, Compose, Execute).

PSEUDO ITALIC ON/OFF (8400/8600 only)

To activate the default value on the 8600* or 8400 or the redefined value of either form of pseudo italic on the 8400, press UDK and the lower case "v". This produces **<PI** and a pause for O for ON or X for OFF. Type "O" and the close Control symbol will appear: **<PIO>**. From that point on characters will be typeset in the pseudo italic style. Pseudo italic can be turned on and off or redefined within a line.

To deactivate, press UDK, type "v" and X to produce **<PIX>**.

NOTE: The 8600 has a default value of 12-degrees Right only. No changes in degree or direction of slant will be recognized for that photo unit. Pseudo Italic can be turned ON and OFF only.

Reverse Video

REVERSE VIDEO

When this function is activated, it causes the 8400 or 8600 to reverse the stroking sequence of characters. In normal operation the beam which generates characters is turned on and off to create character images. In Reverse Video, the background or outline of characters is stroked rather than the character itself and produces a white image on a black background. Word spaces, fixed spaces and letter-spacing also appear in negative form.

Several minor changes in keyboard procedures are necessary when using Reverse Video. If an entire quadded line is to be typeset in Reverse Video, an Insert Space must be entered where reverse blank space is to appear. For example, instead of ending a line with Quad Left, Return, you must use Insert Space, Return. Centered lines must have an Insert Space on either side of the line.

If used for material where extra space is to appear between lines, that space must be included in the line space. If Add Lead codes are used, that amount of white space will appear between lines.

On a line-to-line basis, the depth of Reverse Video strokes will default to the leading value or the largest point size in the line, whichever is greater. To prevent clipping the bottom of Reverse Video characters, leading should always be equal to or greater than the largest point size in the line which follows.

Certain functions have limited use with Reverse Video since they could cause partial or total overprinting of characters. These include: Flash Only, Space Only, End Line, Back Unit/Point, Character Compensation, Negative Letterspacing, Plus/Minus Point or Line Space, Reverse Lead and Pseudo Italic. Reverse Video should not be used with italic type faces, ornamental characters or accented characters for the same reason.

Positive Video

a b c abc

Reverse Video

a b c abc

Centered Line

Sample Line

With Insert Spaces

Sample Line

Reverse Video Command Structure

Compugraphic Corporation

<PS18<RVO>Compugraphic<RVX> Corporation<QC>Return

<ALD6> Between Lines

Reverse
Video

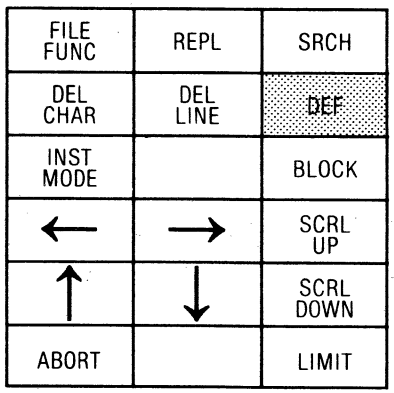
<LS24>

Reverse
Video

User Define Keys

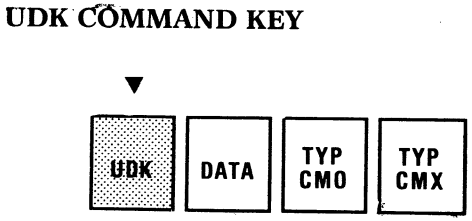
USER DEFINE KEYS

A standard feature of the MCS provides the ability to program certain alphanumeric keys on the typewriter keyboard. This is achieved by the operator assigning text material and/or function commands to individual keys for instant recall. The shift and unshift position of affected keys can be programmed for different conditions. UDK's can be used to eliminate keystrokes that would otherwise be required for repetitive data. Stored data is recalled instantly by the designated keys. Each UDK is activated by either one or two keystrokes. Contents of the key are inserted at the position of the cursor in both the Insert and overstrike modes.



SINGLE KEYSTROKE UDK'S

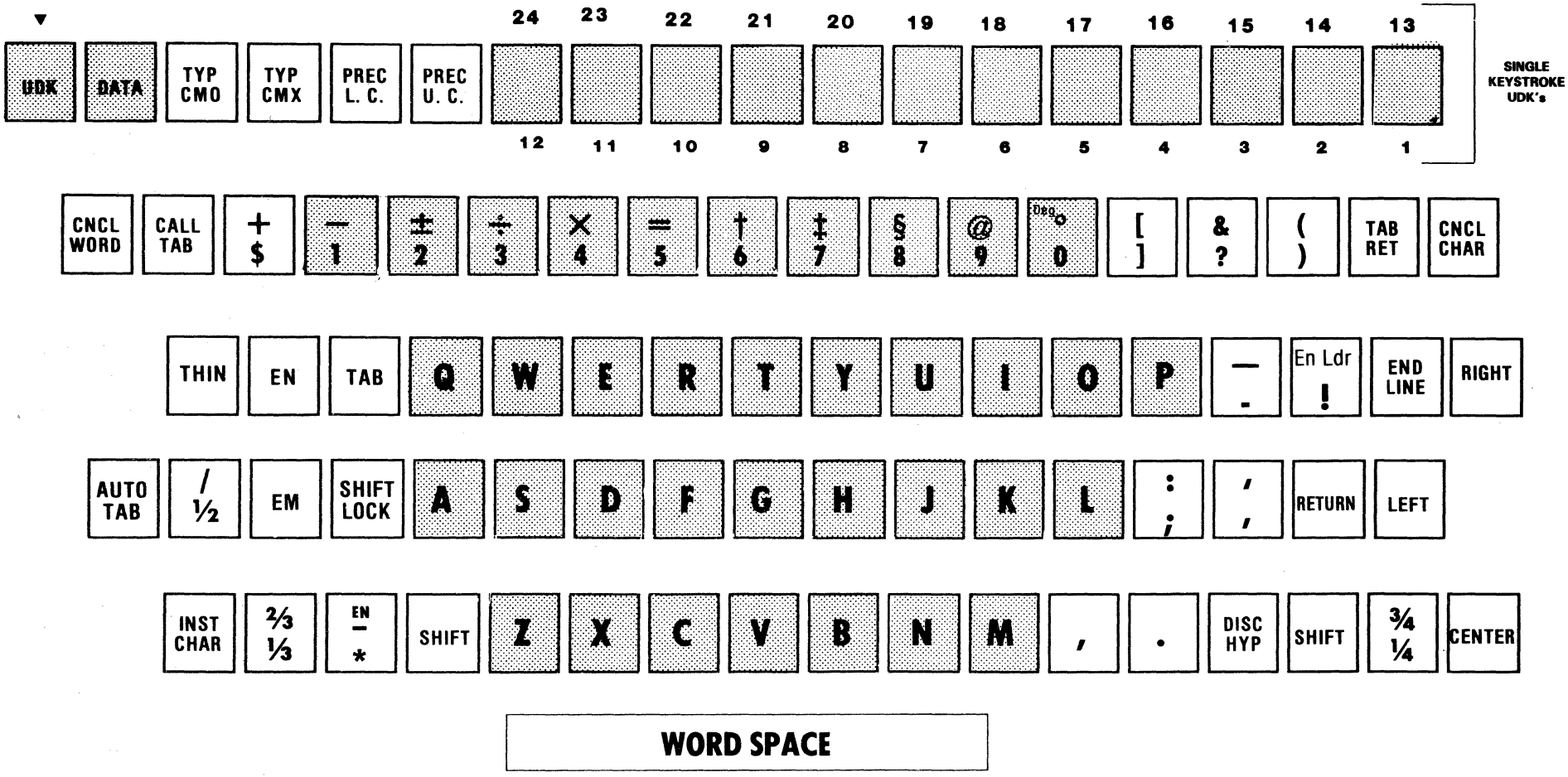
Single-key UDK's are the twelve rightmost keys on the top row, upper and lower case, for a total of 24 positions. Identification numbers are shown on page 87. To store data, press the Define key (editing keypad) and the selected UDK (key numbers 1 to 12 lower case; 13 to 24 upper case). The UDK command key is at the leftmost position on the top row. After data has been entered, the function is completed with *Execute*. When so defined, UDK's are stored *temporarily* and must be redefined whenever the system is re-programmed. If a UDK is to have permanent status, it must be stored in a file on the disk, as described on page 93.



TWO KEYSTROKE UDK'S

Two-keystroke UDK's are designated by the Define Key followed by the UDK command key and the appropriate alphanumeric character. They are accessed by the UDK command key followed by the designated alphanumeric character in the proper shift or unshift position. Stored data will be inserted directly ahead of the cursor.

PROGRAMMABLE USER DEFINE KEYS



UDK COMMANDS

Define UDK (Single Key) — Define/Numeric Character/Enter Data/Execute
Define UDK (Two-Key) — Define/UDK Command/Alphanumeric Character/Enter Data/Execute
Display Single Key UDK — Display Command/Numeric Character
Display Two-Key UDK — Display Command/UDK Command/ Alphanumeric Character
Recall UDK (Single Key) — Press UDK numeric Key
Recall UDK (Two-Key) — UDK Command/Defined Alphanumeric Character
Load UDK File — System Command/LUDK/File Name/Execute
Clear UDK — Define/UDK/Execute

User Define Keys

DEFAULT UDK'S

As an aid to the operator, the system automatically stores commands in the User Define Keys. This default data is entered into both the one- and two-keystroke UDK's when the system is programmed (see page 89). It should be noted that some functions are programmed lower case while other commands with related functions are programmed in both upper and lower case positions of the defined character. The total number or characters used in the default data is 434, allowing 566 characters to be defined as UDK's. As the characters are entered they are counted, with the positions remaining being indicated on message line two.

It is suggested that the typewriter numeric positions (1 to 0) be used for temporary UDK's. Entering them by number will aid the operator in remembering the UDK's used for temporary data. A list of the default UDK's and their defined functions are shown on the following page. The two-key UDK's are accessed by pressing the UDK command key followed by the appropriate character key.

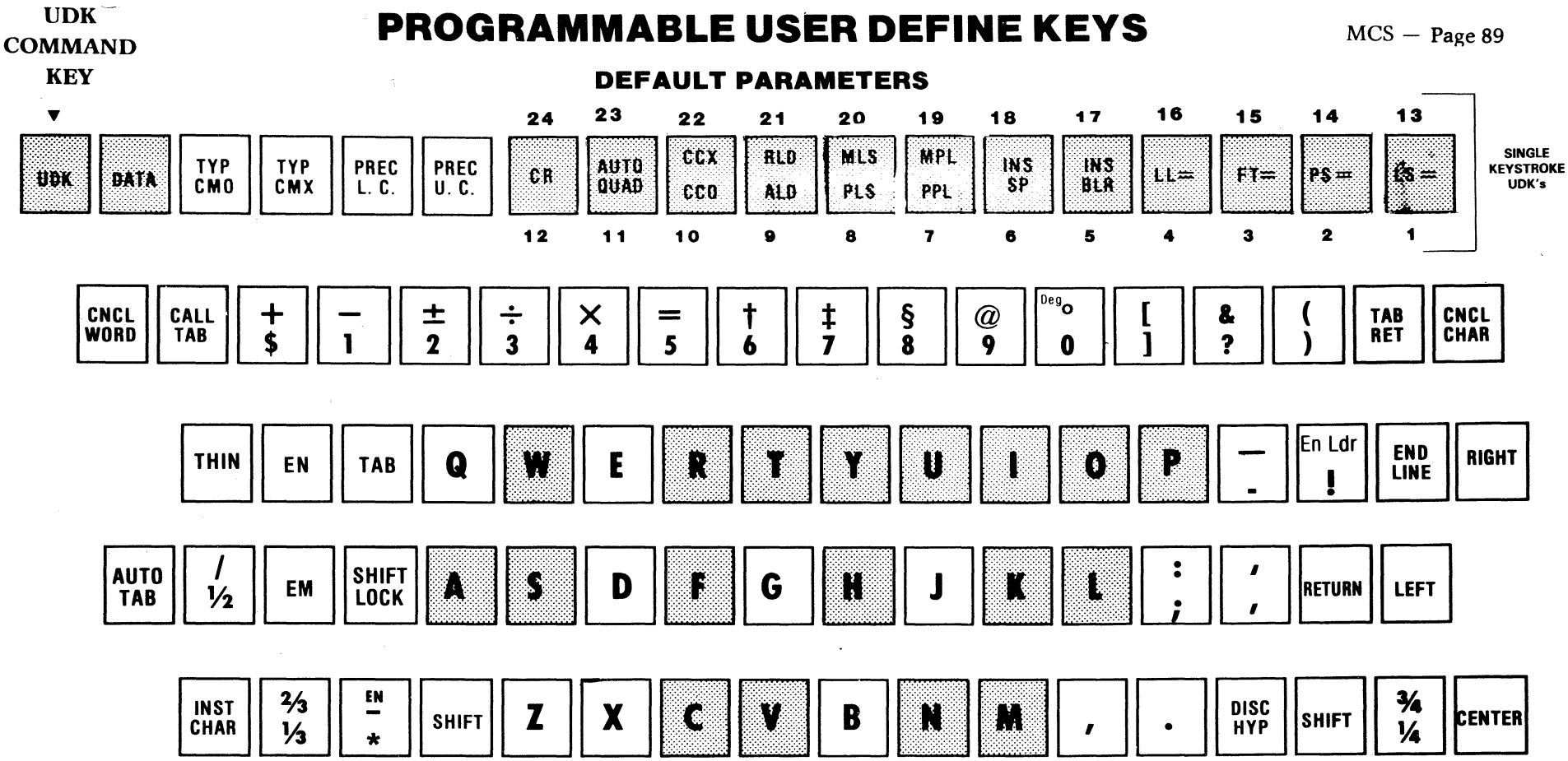
LIMITATIONS

Data can be stored in UDK's as required but is limited to a total of *1,000 characters* for the combined number of UDK's. Invalid entries will produce a brief audio tone and the message "Bad UDK Id". If 1,000 characters are used, further entry will be aborted and the message "UDK overflow" will be displayed. Whenever overflow occurs, data up to that point can be saved by pressing *Execute*. During data entry, errors may be cancelled. However, once defined, a UDK cannot be aborted and must be redefined.

DEFAULT UDK'S (Single Key)
No.
1 Line Space*
2 Point Size*
3 Font*
4 Line Length*
5 Insert Baseline Rule*
6 Insert Space*
7 Plus Point Leading
8 Plus Line Space
9 Advance Leading
10 Character Compensation On
11 Automatic Quadding*
12 Carriage Reset*
19 Minus Point Leading
20 Minus Line Space
21 Reverse Leading
22 Character Compensation Off

DEFAULT UDK'S (Two-Key)
Alphanumeric Characters
w Set Size*
t Define Tab
T Define Floating Tab
u Forward Unit
U Back Unit
i Indent On/Off
I Indent H, L, R, B
o Cut Left
O Cut Right
p Forward Point
P Back Point
a Line Ending (A, S, M)*
s Spaceband*
f Flash Only*
h Hyphenation On
H Hyphenation Off
l Letterspace On/Off
L Letterspace
c Character Compensation Set
C Auto Compensation Select
v Pseudo Italic On/Off
V Pseudo Italic
n No Flash On
N No Flash Off
k Kerning On
K Kerning Off
m Multi Column On/Off/Return
M Multi Column Define
***Upper and Lower Case**

PROGRAMMABLE USER DEFINE KEYS


UDK COMMAND KEY
DEFAULT PARAMETERS
24 23 22 21 20 19 18 17 16 15 14 13
UDK DATA TYP CMO TYP CMX PREC L. C. PREC U. C. CR AUTO QUAD CCX CCO RLD ALD MLS PLS MPL PPL INS SP INS BLR LL= FT= PS= LS=
12 11 10 9 8 7 6 5 4 3 2 1
SINGLE KEYSTROKE UDK's
CNCL WORD CALL TAB + $ — 1 ± 2 ÷ 3 × 4 = 5 † 6 ‡ 7 § 8 @ 9 Deg° 0 [ ] & ? ( ) TAB RET CNCL CHAR
THIN EN TAB Q W E R T Y U I O P — - En Ldr ! END LINE RIGHT
AUTO TAB / ½ EM SHIFT LOCK A S D F G H J K L : ; ' ' RETURN LEFT
INST CHAR ⅔ ⅓ EN — * SHIFT Z X C V B N M , . DISC HYP SHIFT ¾ ¼ CENTER

DEFAULT UDK'S (Single Key)

No.

1 <LSxxxxx> Line Space*
2 <PSxxxx> Point Size*
3 <FTxxxx> Font*
4 <LLxxxxx> Line Length*
5 IC/BL - Insert BLR*
6 <IS> Insert Space*
7 <PPL> Plus Point Leading
8 <PLS> Plus Line Space
9 <ALDxxxxxx> Advance Leading
10 <CCO> Character Compensation On
11 <Ax> Automatic Quadding*
12 CR - Carriage Reset*
19 <MPL> Minus Point Leading
20 <MLS> Minus Line Space
21 <RLDxxxxxx> Reverse Leading
22 <CCX> Character Compensation Off

DEFAULT UDK'S (Two-Key)

Alphanumeric Characters

w <SSxxxx> Set Size*
t <TBxx,xxxxx,xxxxx,x> Define Tab
T <TFxx,xxxxx,x> Define Floating Tab
u <FUxx> Forward Unit
U <BUxx> Back Unit
i <Ix> Indent On/Off
I <Ixxxxxx> Indent H, L, R, B
o <CLx,xxxxxx,xxxxxx,xxxxx> Cut Left
O <CRx,xxxxxx,xxxxxx,xxxxx> Cut Right
p <FPxx> Forward Point
P <BPxx> Back Point
a <xLE> Line Ending (A, S, M)*
s <SBxx,xx,xx> Spaceband*
f <FO> Flash Only*

DEFAULT UDK'S (Two-Key)

Alphanumeric Characters

h <HYO> Hyphenation On
H <HYX> Hyphenation Off
l <LTx> Letterspace On/Off
L <LTxx.xx> Letterspace Set: Positive, Negative
c <CCxx> Character Compensation Set
C <ACSx> Auto. Compensation Select
v <PIx> Pseudo Italic On/Off
V <PIxx> Pseudo Italic Degree
n <NVO> No Flash On
N <NFX> No Flash Off
m <MCx> Multi Column On/Off/Return to Reference
M <MCOxxxxx.xx.xx> Define Multi Column

*Function duplicated in Upper and Lower Case.

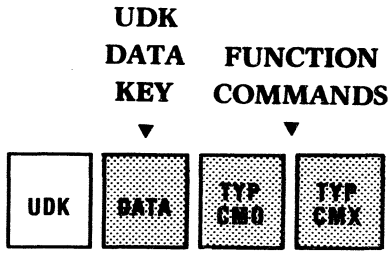
NOTE: If Carriage Reset is to be moved to another key location, it must be redefined for the UDK by Precedence and Return. This produces the special CR logo on the screen.

DATA ENTRY

Data entry for commands must be complete, including the Open (CMD) and Close (CMX) [< >] function commands. The parameter values, however, can be substituted by entering a special code with the data key. *The special data code, indicated by a symbol, allows for variable data entry, for function commands where parameter values will differ.* For example, commands for Point Size, Line Space, Type Fonts, Tabs, etc. When the UDK is recalled, all data, up to the special data code will be input to the videoscreen, indicated with the message "ARG?" (Argument). The system will then pause until the operator completes the entire command with the required parameters.

STRINGING COMMANDS

When defining commands, both the Opening (<)and Closing (>) command codes are required. However, additional keystrokes and data codes can be saved by stringing commands. This is achieved by combining data commands wherein only the Open command code is entered. The Close command is entered at the end of the command string by the direct entry of the Close (>) command.



VARIABLE DATA SYMBOLS, POINT SIZE SELECTION

<PS0000>

DATA SYMBOLS

UDK STRING COMMAND

<LL22.10<PS10<FT2<LS12.2>

User Define Keys

DEFINING UDK's

UDK's can be defined or redefined at any time and will override any data previously stored in the same locations. An individual UDK can be defined for temporary use or a permanent UDK file can be created for daily use. When programming an individual UDK, data is displayed on the top message line. Errors can be deleted with the Cancel Character key. The key being defined and the number of characters available are displayed in the second message line. In the example below, the "x" key is being defined and there are 509 undefined characters remaining.

Display of data being loaded:

UDK .x 0509 I 1657

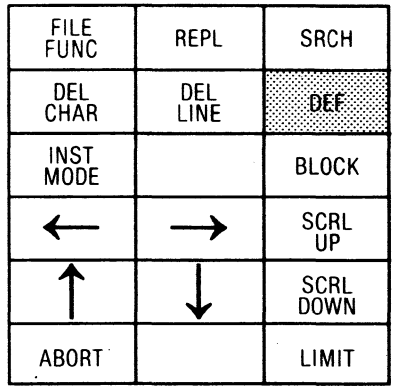
Regardless of how many characters are stored in a single UDK, only 52 characters can be displayed on the top message line. When defining a UDK, if more than 52 characters have been entered, cancelling one character moves another back onto the display. Redefining an individual UDK does not affect the UDK file. However, no permanent record is made of the redefined UDK unless it is stored in a UDK file.

Loading UDK's

A single UDK is loaded by pressing *Define,* on the Editing keypad, followed by the UDK key, the key where data is to be stored, then the data itself. *Execute* completes the command. Several UDK's can be loaded by entering *Define* instead of *Execute* at the end of all but the last definition.

Clear UDK's

Data in a UDK can be deleted by defining the key without entering any data. The key sequence *Define/UDK/Execute* clears data from the key addressed. Entry of new data also erases any previously stored in that location.



Recall

A single-key UDK (such as LL, FT, Call Tab, etc.) is recalled by pressing the appropriate key. UDK data is always inserted at the cursor location and will not delete any characters.

To recall a two-key UDK, press the UDK command key and the appropriate alphanumeric key.

As recalled data is entered, the system looks for the variable data symbol. If one is found, a pause occurs for the operator to supply the necessary keystrokes. If none is encountered, the function is complete. Data will be counted and composed if the Compose mode is active.

DEFINING UDK FILE

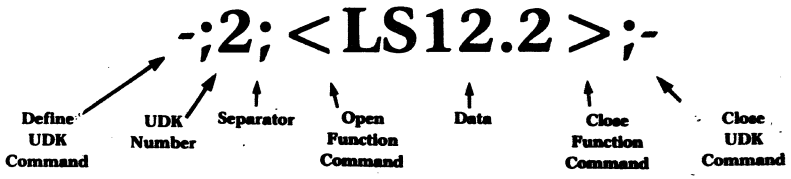
UDK's can be completely and quickly re-defined by establishing UDK files by name and storing them on the user disk. In order to define UDK's in a file, a specific code structure is required. It consists of opening and closing commands for the individual UDK's. The codes take the place of the Define and UDK command keys when establishing the UDK's directly from the keyboard. Whenever the special data codes are involved, *the Edit Mode of operation should be invoked* **(File Function/EDIT).**

The opening codes for identifying a UDK are the hyphen followed by a semicolon (-;). The closing codes are the semicolon followed by a hyphen (;-). The command varies for the single- and two-key UDK's. The single-key command is identified by the opening command followed by the number of the UDK (See Keyboard A). A semicolon is then entered to separate the identity of the UDK from the required text or functional data (-;3;data;-).

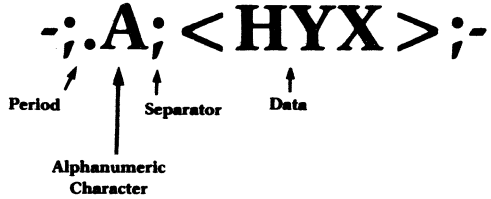
A two-key file command consists of the opening codes followed by a period (.) and the alphanumeric character for the two-key UDK (-;.A). A semicolon must be entered to separate the UDK identity from the text data (-;.A;data;-).

After the UDK alphanumeric character is defined, the text data is entered. The data can include any alphanumeric characters and/or function codes. If the data is functional, it must include the opening (<) and closing (>) typesetting command codes. The UDK definition is then completed with the close UDK command, i.e., a semicolon followed by a hyphen (;-). When a semicolon is used in data output, an additional semicolon must be entered to distinguish it from the closing function. (;;). If a UDK file function is entered in error, the defined UDK will be ignored. When called in error, the UDK will be indicated with the message, Undefined

DEFINED SINGLE KEY UDK
Line Space, 12½ Points



DEFINED TWO-KEY UDK
Hyphenation Off



DEFINED TWO-KEY UDK
Text Data

-;.Q;Compugraphic←;-

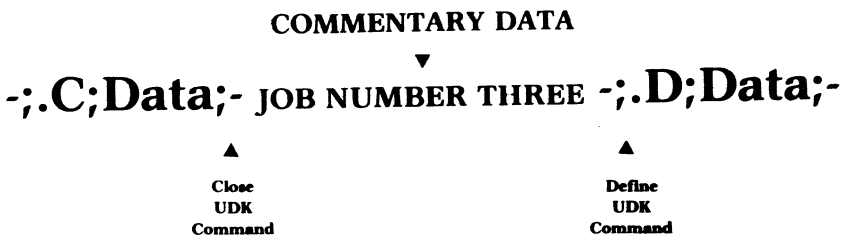
VARIABLE DATA, LINE SPACE PARAMETERS

-;2;<LS0000>;-

DATA SYMBOLS

Redefining a UDK stored in memory has no affect on the file. The data stored in a re-defined UDK cannot be saved and stored in the file. If the re-defined UDK data must be saved, it has to be entered into a UDK file.

It should be noted that any data entered between the end of one UDK (;-) and the beginning of the next UDK (-;) will be automatically deleted when the file is loaded. This area can be used to enter comments or reminders within the file(s).



LOADING UDK FILES

After the creation of a UDK file, it must be loaded into memory. The file is loaded with the System Command function located on the File Management Control Keypad (See System Commands, File Management Reference). The System Command is followed by the mnemonic data abbreviation "LUDK" (Load UDK). The file name followed by *Execute* stores the file in the UDK memory. The data will be entered sequentially while rapidly being displayed in the message area. *Caution: Whenever a UDK file is loaded, all data previously stored in the UDK memory is deleted.*

SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	

LOAD UDK FILE

Procedure: *System Command*/LUDK/File Name/*Execute.*

MERGE UDK FILE

In order to preserve UDK data stored in the keyboard memory, a supplementary UDK file can be merged with data currently stored in memory. However, merged data will override any data already stored in a duplicated key position. The UDK file is merged rather than loaded since the latter procedure erases all data stored in UDK's.

The sequence *System Command / M / U / D / K /* File Name / Disk Drive/*Execute* is used to merge the UDK file with data stored in the keyboard memory.

DISPLAY UDK

For the purpose of verification, data currently stored in a UDK can be displayed on the top message line. If more than 52 characters are stored in a key, the entire string will step across the screen and stop with the last 52 displayed.

The sequence *DISPLAY key/UDK key/alphanumeric key* will display the contents of any defined key on the message line.

No corrections can be made in the displayed data. If any errors are detected, the key must be reprogrammed.

MERGE UDK

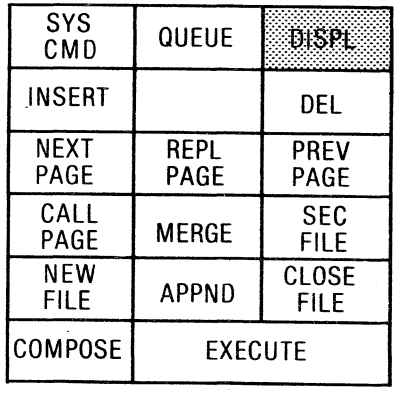
Procedure: *System Command* / M U D K / File Name / Disk Drive / *Execute*

DISPLAY UDK (2-Key)

Procedure: *Display / UDK Key* / Alphanumeric Character

DISPLAY UDK (1-Key)

Procedure: *Display / Single-Key UDK*



SAMPLE UDK FILE

-;1;<LS12>;- (Line Space)
-;2;<PS10>;- (Point Size)
-;3;<FT9>;- (Font 9)
-;4;<LL27.6>;- (Line Length)
-;5;IC/BLR;- (Insert Base Line Rule)
-;6;<IS>;- (Insert Space)
-;7;<PLS>;- (Plus Line Space)
-;8;<MLS>;- (Minus Line Space)
-;9;<CC2>;- (Character Compensation 2 Units)
-;10;<CCO>;- (Character Compensation On)
-;11;<CCX>;- (CharacterCompenstion Off)
-;12;<FO>;- (Flash Only)
-;.H;<HYX>;- (Hyphenation Off)
-;.h;<HYO>;- (Hyphenation On)

CHANGE DISPLAY MAPPING

Four keyboard arrangements are available for display of characters on the screen: 500, 501, 549 and 1614. Two arrangements can be accessed with each User Program Disk but only one can be in use at a time. Complements are selected when creating a User Program Disk. The answer to the System Generator prompt: *What is the Base Display Map?* identifies the Map 1 complement. If 000500 is specified, the 500 complement becomes Map 1 and is in effect at the start of each new job.

The answer to *What is the alternate Display Map?* becomes the Map 2 complement. Several User Program Disks can be created so that different pairs of complements can be accessed. Although only one mapping table is active at a time, fonts with any of the four complements can be addressed. The only limitation is that the characters displayed for certain typewriter key positions will not match the characters that will be typeset. Images for alphanumerics and basic punctuation in all four complements are displayed correctly regardless of which mapping table is in use.

The character set is changed with the *System Command* key, followed by the letters D M A P. This displays the current complement number on the top message line. Type the new complement number and press *Execute.* If you decide not to change the complement, the function can be aborted prior to its execution.

CHANGING DISPLAY MAPPING

Procedure: *System Command* / D M A P / 1 / *Execute* or
System Command / D M A P / 2 / *Execute*

500 CHARACTER COMPLEMENT

★ ✓ - ® • □ % / ¢ _
1234567890
$1234567890[&(
$1234567890]?)
QWERTYUIOP— .
qwertyuiop-!
/ASDFGHJKL:'
½ asdfghjkl;'
⅔ ¢ZXCVBNM,. ¾
⅓ *zxcvbnm,. ¼

501 CHARACTER COMPLEMENT

★ ✓ # ® © □ • % ● _
☆ { } | ¶ ■ ◄ ► ' "
+ − ± ÷ × = † ‡ $ @ ° [&(
$1234567890]?)
QWERTYUIOP— .
qwertyuiop-!
/ASDFGHJKL:'
½ asdfghjkl;'
⅔ –ZXCVBNM,. ¾
⅓ *zxcvbnm,. ¼

549 CHARACTER COMPLEMENT

¡ / * ¢ © ~ • Ç % _
! º ª # ® ~ . ç « »
+ − ± ÷ × = † ‡ - @ ° & ¿ (
$1234567890é?)
QWERTYUIOP — ß
qwertyuiop-ı
` ASDFGHJKL:'
` asdfghjkl;'
^ ´ ZXCVBNM ,. ¨
^ ´ zxcvbnm ,. ¨

1614 CHARACTER COMPLEMENT

□ ffiffl ¡ ´ ` ^ ¨ ~ _
£ ıß ¿ ´ ` ^ ¨ ~ ¸
¢ 1 fiflfff # % ⅛ ⅜ ⅝ ⅞ [&(
$1234567890]?)
QWERTYUIOP— .
qwertyuiop-!
/ASDFGHJKL:'
½ asdfghjkl;'
⅔ •ZXCVBNM,. ¾
⅓ *zxcvbnm,. ¼

SENDING DATA TO BE TYPESET

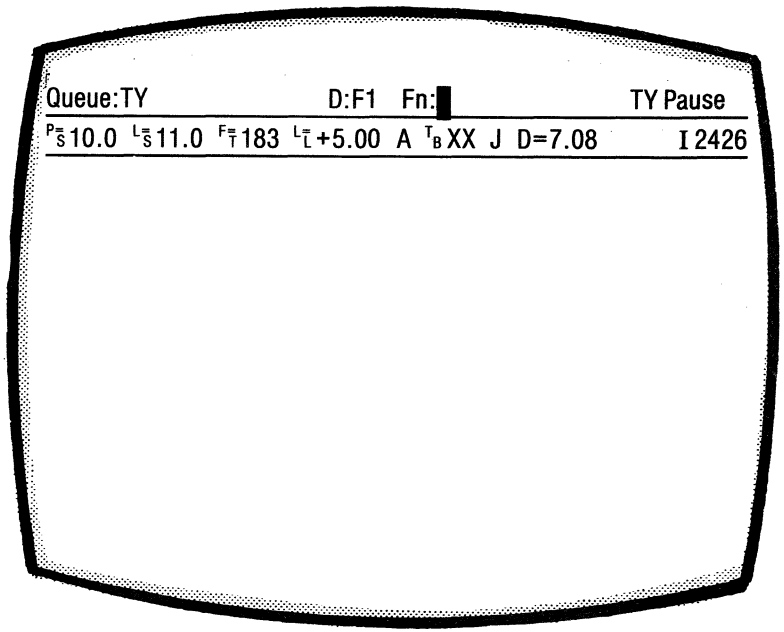
On completion of a job and after the file has been closed, data can be sent to the photo unit to be typeset. Pressing *Queue* on the File Management keypad followed by the mnemonic codes "TY," the file name and Execute sends the file.

The message line will display TY Job and the job number, which represents the number of jobs that have been sent to the photo unit since the system was last programmed. Any number of jobs can be sent in quick succession and each will have an incremented job number.

The status of the queue can be examined by *Display/TY/Execute.* This produces an index of jobs currently in the queue. A job number is listed in the index beside each file name. Also displayed are the following status indicators.

COMPOSING:	The job is being recomposed in the background before being sent to the photo unit.
WAITING:	The job has been logged on to the queue but has not yet been recomposed.
TYPESETTING:	The job is being typeset.
COMPLETED:	The job has been typeset.
ABORTED:	The job was not typeset and must be entered in the queue again.

SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	



SYSTEM COMMAND RESET

The System Command *Reset* resets the typesetter and clears the typesetter queue.

Procedure: *System Command/RES/TY/Execute*

Typesetter Queue

DELETING JOBS FROM THE TYPESETTER QUEUE

The TY queue should be examined occasionally to check on the status of jobs sent to the photo unit. Jobs which are typeset routinely are dropped from the queue. If a typesetting problem occurs, the file name will remain in the TY queue and an error message will be displayed in the right-hand column. This can be caused by a font not having been loaded in the terminal or photo unit, by a depleted paper supply or by the photo unit not being on-line. An "aborted" message in the TY display indicates that the file was not typeset.

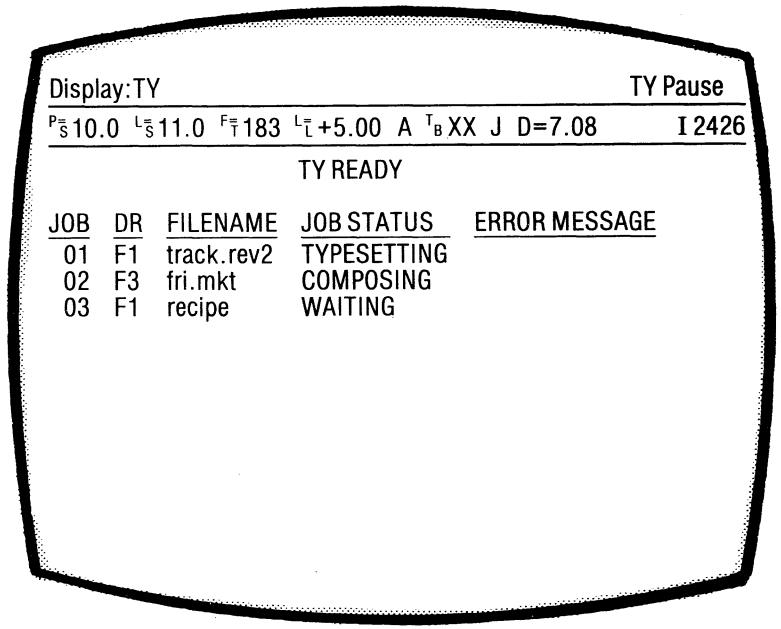
With the TY queue displayed, make a list of any such files and the nature of the problem. Delete the files from the TY queue by job number:

System Command/D J O B/T Y/job number/*Execute*

The queue listing does not update until the TY queue is redisplayed. After correcting any problems, jobs can be sent to the photo unit again.

TY MESSAGES	DESCRIPTION
No Film	Photo unit paper supply depleted.
Fatal Error	Photo unit could not process file.
## Abort	Job ## aborted; illegal command.
## Data	Unimplemented command.
	Invalid character used in parameter.
## Font	Invalid font.
No Font	Font library invalid.
No Cassette	Cassette not loaded; not engaged properly.
Error	Typesetter not ready.
## Disk	Disk error (Read, Write, No File or Directory).
No Device	Typesetter not on-line.
Pause	Typesetter in pause state.
Com Error	Communications error.

SYS CMD	QUEUE	DISPL
INSERT		DEL
NEXT PAGE	REPL PAGE	PREV PAGE
CALL PAGE	MERGE	SEC FILE
NEW FILE	APPND	CLOSE FILE
COMPOSE	EXECUTE	


Display:TY
TY Pause
P=S 10.0 L=S 11.0 F=T 183 L=L +5.00 A TB XX J D=7.08
I 2426
TY READY
JOB DR FILENAME JOB STATUS ERROR MESSAGE
01 F1 track.rev2 TYPESETTING
02 F3 fri.mkt COMPOSING
03 F1 recipe WAITING

AUTOMATIC KERNING

Kerning is the reduction of white space between letter combinations. With the standard program, it can be performed manually by placing Back Unit (or Back Point) commands between letter combinations as they occur. This improves typographic appearance but requires operator participation.

Two kerning programs are available for MCS-compatible fonts: Standard Kerning and Designer Kerning. All width disks manufactured for MCS systems after Sept. 1, 1982 contain Standard Kerning and AVCC program information. Standard Kerning is most effective when used with text sizes, 12 point and under. It is also effective in larger point sizes when combined with AVCC. Designer Kerning is an option and must be ordered from the Type Division. It is more effective for display sizes, 14 point and up. Customers who wish to order Designer Kerning, or add Standard or Designer Kerning to font libraries established before September, must order new width disks for the fonts involved from the Type Division. Both styles also include AVCC, which can be used independently or with either kerning program.

Both kerning programs automatically reduce space between pre-programmed letter combinations as they are typed. *Letters must be adjacent for kerning to apply; no function symbols can intervene.* Kerning is off at the start of each new job. It is activated with UDK/k, which produces <**KO**>. Kerning takes place automatically in the background with no operator involvement. No codes appear between letters on the screen.

Kerning is deactivated by UDK/K which produces <**KX**>. It can be turned on and off at any valid point in a line. Kerning should not be used with Negative Video or with Script faces where clipping could occur between kerned letters.

EXAMPLES OF KERNED LETTER COMBINATIONS

(Kerning OFF, First Line; Kerning ON, Second Line)

To Wa Va PA y.
To Wa Va PA y.

v. p, ay VA AT
v. p, ay VA AT

AW LT Yo Fr Fo
AW LT Yo Fr Fo

ya xe te we ro
ya xe te we ro

When an index of fonts loaded in the terminal is displayed, the right-hand screen column identifies fonts which have kerning capability. Standard Kerning is identified by "T" or "TX"; Designer Kerning is identified by "D" or "DX". The letter "X" indicates that customer alterations have been made to the values supplied by the Type Division. A blank space in the column indicates that the font does not have kerning.

Automatic Variable Character Compensation

AUTOMATIC VARIABLE CHARACTER COMPENSATION

AVCC is incorporated into the Automatic Kerning option. It consists of three graduated character compensation programs entitled *Book Set, Ad Set* and *Tight Set.* Special WIDTH disks prepared by the Type Division to provide Kerning values also contain values for all three levels of compensation. The master width disks are used to create User Width Disks, which in turn are used daily to load widths into the terminal.

Book Set is activated by **<ACS1<ACO>** and causes a slight decrease in white space between every letter combination.

Ad Set is activated by **<ACS2<ACO>** and results in an intermediate decrease in white space between letters.

Tight Set is activated by **<ACS3<ACO>** and produces the tightest possible character fit.

When any of the three factors is activated, the message line will display **CC = xx A** with xx's representing the compensation factor in ninth-of-a-point increments. The letter "A" indicates an AVCC factor rather than one associated with the basic character compensation program, which would be indicated by "M". Current status can also be verified by calling up the Parameter Display Index. For example, column one "Char. Comp. Set: O 3,02" would indicate that a *Tight Set* factor of 2/9ths of a point was active.

If AVCC was turned on but no style was selected, *Book Set* would be in effect. Once activated, a style can be turned on and off, or a different style can be selected. The compensation factor increases or decreases as the point (or set) size is changed. AVCC is turned off by **<ACX>**.

When an index of fonts loaded in the terminal is displayed, *Display*/FT/*Execute,* the letter "Y" in the fifth column identifies AVCC fonts. Non-AVCC fonts display the letter "N".

NOTE: The *Ad Set* program was used when composing this page.

◄ NINE POINT

NORMAL SET Founded in 1950 as an educational center for the improvement of rea
BOOK SET Founded in 1950 as an educational center for the improvement of rea
AD SET Founded in 1950 as an educational center for the improvement of rea
TIGHT SET Founded in 1950 as an educational center for the improvement of read

◄ TEN POINT

NORMAL SET Founded in 1950 as an educational center for the improvemen
BOOK SET Founded in 1950 as an educational center for the improvemen
AD SET Founded in 1950 as an educational center for the improvemen
TIGHT SET Founded in 1950 as an educational center for the improvement c

◄ ELEVEN POINT

NORMAL SET Founded in 1950 as an educational center for the improv
BOOK SET Founded in 1950 as an educational center for the improv
AD SET Founded in 1950 as an educational center for the improve
TIGHT SET Founded in 1950 as an educational center for the improver

◄ TWELVE POINT

NORMAL SET Founded in 1950 as an educational center for the in
BOOK SET Founded in 1950 as an educational center for the in
AD SET Founded in 1950 as an educational center for the im
TIGHT SET Founded in 1950 as an educational center for the imp

◄ THIRTEEN POINT

NORMAL SET Founded in 1950 as an educational center for th
BOOK SET Founded in 1950 as an educational center for th
AD SET Founded in 1950 as an educational center for th
TIGHT SET Founded in 1950 as an educational center for the

◄ FOURTEEN POINT

NORMAL SET Founded in 1950 as an educational center fo
BOOK SET Founded in 1950 as an educational center fo
AD SET Founded in 1950 as an educational center for
TIGHT SET Founded in 1950 as an educational center for t

NOTE: To select an AVCC style, press UDK/C and a single figure. The current program does not have a key devoted to **ACO/ACX**, so temporarily a UDK key of your choice can be programmed to turn AVCC on and off.

PICA/CICERO/METRIC CONVERSION

The **MCS** domestic program has a basic counting system configured for Picas and Points. The counting system can be temporarily changed however, for calculation in either Cicero or Metric units. The change can be made to reflect either horizontal (line measure, tabs, indents, cuts) or vertical (line space, add line space) counting.

For horizontal calculation the unit value is changed by adding either a "C" (Cicero) or "M" (Metric) after the designated value of the line length command structure. If the counting is to be converted for line space, it too must be entered after the designated value.

The change in calculation is a *temporary change* and must be entered for each command. If the change is not made, counting reverts to Picas and Points. *It should be noted that although the calculation is converted to Cicero or Metrics, it is always represented on the status lines in the basic system, i.e., Picas and Points.*

Line Length: Cicero/Metric

⟨LL12C⟩

⟨LL18.6C⟩ (Can be entered in either upper or lower case)

⟨LL12M⟩ **Millimeters/10ths of Millimeters**

Line Space: Cicero/Metric

⟨LS120C⟩

⟨LS10M⟩

COPY PREPARATION

The Point and Pica are standard units of measurement used in the markup of copy. The Point and Pica are used by the operator to define typesetting parameters for the photo unit.

The Point

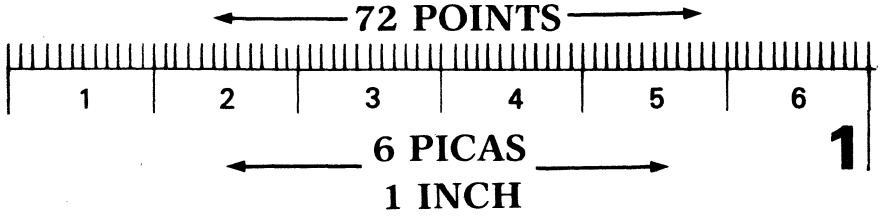
The point equals 1/72nd of an inch. There are 72 points to one inch.

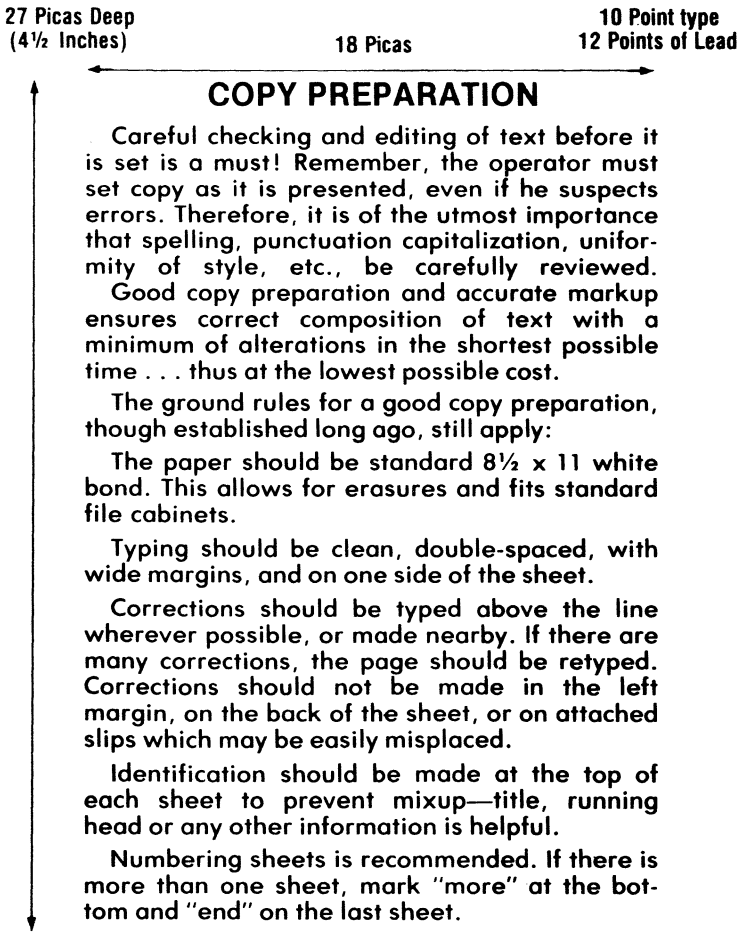
The Pica

The Pica is a larger unit of measurement. The Pica equals 12 points or 1/6th of an inch. There are six picas in one inch.

The line lengths for copy blocks are specified in Picas, as well as the depth. For example, a block of copy three inches wide by four and one-half inches deep would be designated as 18 picas wide and 27 picas deep. The line length is entered as one of the typesetting parameters. The depth count is maintained by the system. The current depth is displayed in picas and points in the second message line.

Type Sizes are referred to as Point Size. Depending on the photo unit, point sizes range from 4 to 72 point. Line Space is indicated in points and quarter-points.




27 Picas Deep
(4½ Inches)
18 Picas
10 Point type
12 Points of Lead
COPY PREPARATION
Careful checking and editing of text before it is set is a must! Remember, the operator must set copy as it is presented, even if he suspects errors. Therefore, it is of the utmost importance that spelling, punctuation capitalization, uniformity of style, etc., be carefully reviewed.
Good copy preparation and accurate markup ensures correct composition of text with a minimum of alterations in the shortest possible time . . . thus at the lowest possible cost.
The ground rules for a good copy preparation, though established long ago, still apply:
The paper should be standard 8½ x 11 white bond. This allows for erasures and fits standard file cabinets.
Typing should be clean, double-spaced, with wide margins, and on one side of the sheet.
Corrections should be typed above the line wherever possible, or made nearby. If there are many corrections, the page should be retyped. Corrections should not be made in the left margin, on the back of the sheet, or on attached slips which may be easily misplaced.
Identification should be made at the top of each sheet to prevent mixup—title, running head or any other information is helpful.
Numbering sheets is recommended. If there is more than one sheet, mark "more" at the bottom and "end" on the last sheet.

LINE SPACE

Similar to single- or double-spacing on a typewriter, Line Space controls the advance of paper or film on the photo unit and creates space for a new line. Where a typewriter generally has three settings for paper advance, Line Space ranges from ¼-point to 99¾ points.

Often referred to as leading (ledding), Line Space is specified in points and quarter-points. When the leading for text is equal to the point size, the type is considered as being set solid. For example, if the leading for 10 point type is designated as 10, it would be set solid and marked as 10 on 10 (10 point type, 10 points of line space). If 10 point type is set with two extra points of space between lines, it is specified as 10 on 12 (10 point type, 12 points of line space).

In photocomposition, spacing for lines is calculated from baseline to baseline of the body of the type (see Example A). The baseline is an imaginary line on which the body of letters rests. Capital letters, figures and ascenders occupy the upper two-thirds of the point size, with the remaining one-third for descenders, including a shoulder of white space. The shoulder provides minimum space between lines when type is set solid.

The point size is the determining factor when selecting line space for text. In solid composition, where additional space is not required, leading will be equal to the point size. Where additional space is needed, the extra amount is added to the point size. When point sizes vary from line to line, if the current line is smaller that the line that follows, line space should be equal to or greater than the larger point size. Leading from a 12 point line to a 36 point line should be at least 36 points.

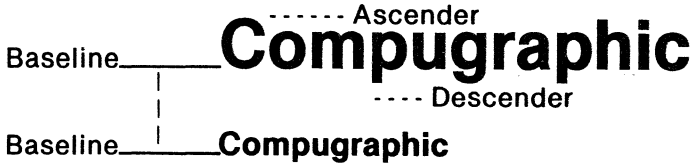
12 POINT TYPE, 12 POINTS OF LEADING

The advantages of modular systems are many. Cost and production savings are the primary factors.

12 POINT TYPE, 16 POINTS OF LEADING

The advantages of modular systems are many. Cost and production savings are the primary factors.

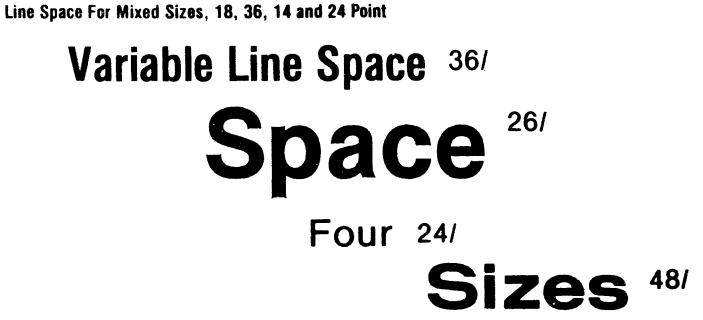
Example: A



However, when the current point size is larger than the line that follows, additional line space must be added to allow for the descenders of the lower case characters of the current line (Example B). The additional space is calculated by allowing one-third of the larger point size to be added to two-thirds of the smaller. This will be the basic line space when mixing most styles of type. For closer fit, minimum space can be calculated by allowing one-third of the large point size and two-thirds of the smaller for the leading value.

When setting all caps, normal leading will prevail. No extra spacing is needed since there are no descenders. Minimum spacing is based on the cap height of the type style and point size. It should be noted that cap height and descenders will vary with some type styles.

Example: B



NOTE: The 48 points of leading after the last line allows extra space for new composition.

MARKUP

Markup of copy is an important aid to the operator. Markup is simple short-hand instructions indicating required specifications to typeset a job. Although markup systems vary, it normally indicates line measures, margins, point sizes, line space and type styles. In addition to markup, a rough layout (illustration) showing the location of the copy elements is recommended.

Markup includes symbols to indicate when copy is to be set small caps or in an italic or a bold style. Markup symbols are also used by proofreaders to show changes and corrections in the typeset copy.

MARKUP SYMBOLS

X21	Represents the Line Length in Picas.
(10)	Represents the Point Size.
F1-16	Represents the assigned type font number.
/12	Represents Line Space (Leading).
12/24	Represents Line Space and additional space between paragraphs or copy blocks.
Compugraphic	Set Italic.
Compugraphic	Set Bold.
COMPUGRAPHIC	Set all caps.
QL	Set copy Flush Left on left margin.
QR	Set copy Flush Right on right margin.
QC	Set copy Quad Center on the line measure.
Just	Set copy justified (margin to margin).

X27 Just.

(10) F3 Markup of copy is an important aid to the operator. Markup is simply 12/24 short-hand instructions indicating required specifications to typeset a job. Although markup systems vary, it normally indicates the line measures, margins, point sizes, line space and type styles. In addition to markup, a rough layout (illustration) showing the juxtaposition of the job is recommended. 12/24

Markup includes symbols indicating when copy is to be set as small caps or in an italic or a bold style. Markup symbols are also used by proofreaders to show changes and corrections in typeset copy.

COPY FITTING

Copy Fitting is the estimation of the number of typeset lines that will fit into a given area. Although Copy Fitting is not always necessary, it is required when a specific style and point size must fit an assigned area.

CHARACTER COUNT

Counting the number of characters in the copy is the most accurate method of Copy Fitting. Because of the many variables—short lines, capital letters, indents, etc., it is an approximate figure.

Most typewriters, having 10 characters per inch, makes character counting quite simple. Drawing a vertical line through the manuscript copy at the width of an average line will determine characters per line. When counting the characters for an average line be sure to include punctuation and word space. Multiplying the average number of characters per line by the number of lines will give the approximate number of characters. For a small volume of manuscript text, the characters outside the drawn margin line should be added to the character count.

PROCEDURES:

Determining Characters For The Manuscript Copy

1. Count the number of typewriter lines.
2. Find the average number of characters for each line by drawing a line through the width of the average line.
3. Multiply the average number of characters per line by the number of lines to find the total number of characters for the manuscript copy.

50 Characters Per Line
× 14 Lines

730 Characters Manuscript Copy **(Includes characters to right of line)**

Drawn Line

MANUSCRIPT COPY →

Drawing a line through the average width of the typewriter copy will determine the average number of characters in a line. Counting the characters in the average line will determine the approximate number of characters in the manuscript copy. Short lines can be counted as full lines, since the typeset copy will also have short lines. If there are several pages of text, it may be necessary to find the average number of lines per page. This is accomplished by adding the total number of lines and dividing by the number of pages. If type style and size are determined in advance, the manuscript copy can be typed with the proper number of characters to the line so that the type setting will run approximately line for line.

TYPEWRITER COPY TO TYPE

To find the number of typeset lines for the manuscript copy, the number of characters that will fit into a pica for the given style and point size is required. The result of multiplying the characters-per-pica* by the pica line length equals the average number of characters per line. Divide the total number of manuscript characters by the number of characters per line to determine the number of lines that will be typeset with those parameters.

PROCEDURES:

Determining the Number of Characters Per Line for a Specified Type Style and Point Size

1. Refer to Typeface Chart (next page) to determine characters-per-pica for the type style and point size (9-point Garth). Round off to the nearest decimal value.
2. Multiply the characters-per-pica by the line length; (example, 18 picas). This will indicate the number of characters per line for the selected style and point size.
3. Divide the total number of manuscript characters by the number of characters per line in the selected style to determine the total number of 18-pica lines.

***The number of characters per pica for the selected type style and point size can be obtained from the Compugraphic Font Division Portfolio of Type, Part No. 55377-0301.**

50 Characters Per Line
× 14 Lines

730 Characters
Manuscript Copy

2.85 Characters Per Pica (from chart)

18-Pica Line Length
× 2.85

51.3 or 52 Characters Per line

14 Lines, 9-Point Garth, 18-pica measure

52 ⌊730 Manuscript Characters

TYPEFACE NAME	CHARACTER PER PICA														
	6	7	8	9	10	11	12	14	18	24	30	36	48	60	72
Garamond Halbfett T/D	4.32	3.71	3.24	2.88	2.59	2.36	2.16	1.85	1.44	1.08	.86	.72	.54	.43	.36
Garamond Kursiv Halbfett T/D	4.55	3.90	3.41	3.03	2.73	2.48	2.27	1.95	1.52	1.14	.91	.76	.57	.45	.38
Garamond No. 2 D	–	–	–	–	–	–	–	2.36	1.84	1.38	1.11	.92	.69	.56	.46
Garamond Italic No. 2 D	–	–	–	–	–	–	–	2.32	1.80	1.35	1.09	.90	.67	.54	.45
Garamond Bold No. 2 D	–	–	–	–	–	–	–	2.14	1.66	1.25	1.00	.83	.62	.50	.42
Garamond Bold Italic No. 2 D	–	–	–	–	–	–	–	2.14	1.66	1.25	1.00	.83	.62	.50	.42
Garamond No. 49 T	–	4.04	3.53	3.14	2.82	2.57	2.35	2.01	1.56	1.17	–	–	–	–	–
Garamond Italic No. 49 T	–	4.04	3.53	3.14	2.82	2.57	2.35	2.01	1.56	1.17	–	–	–	–	–
Garamond Bold No. 49 T	–	4.04	3.53	3.14	2.82	2.57	2.35	2.01	1.56	1.17	–	–	–	–	–
Garamond Bold Italic No. 49 T	–	4.04	3.53	3.14	2.82	2.57	2.35	2.01	1.56	1.17	–	–	–	–	–
Garamond Extrabold No. 49 T	–	3.97	3.47	3.09	2.78	2.53	2.31	1.98	1.54	1.15	–	–	–	–	–
Garamond Extrabold Italic No. 49 T	–	3.97	3.47	3.09	2.78	2.53	2.31	1.98	1.54	1.15	–	–	–	–	–
Garamond Light (ITC) T/D	4.25	3.65	3.19	2.84	2.55	2.32	2.13	1.82	1.42	1.06	.85	.71	.53	.43	.35
Garamond Light Italic (ITC) T/D	4.27	3.66	3.20	2.85	2.56	2.33	2.13	1.83	1.42	1.07	.85	.71	.53	.43	.36
Garamond Book (ITC) T/D	4.23	3.62	3.17	2.82	2.54	2.31	2.11	1.81	1.41	1.06	.85	.70	.53	.42	.35
Garamond Book Italic (ITC) T/D	4.18	3.58	3.13	2.79	2.51	2.28	2.09	1.79	1.39	1.04	.84	.70	.52	.42	.35
Garamond Bold (ITC) T/D	3.77	3.24	2.83	2.52	2.27	2.06	1.89	1.62	1.26	.94	.76	.63	.47	.38	.31
Garamond Bold Italic (ITC) T/D	3.75	3.22	2.81	2.50	2.25	2.05	1.88	1.61	1.25	.94	.75	.63	.47	.38	.31
Garamond Ultra (ITC) T/D	3.54	3.03	2.65	2.36	2.12	1.93	1.77	1.52	1.18	.88	.71	.59	.44	.35	.29
Garamond Ultra Italic (ITC) T/D	3.46	2.97	2.60	2.31	2.08	1.89	1.73	1.48	1.15	.87	.69	.58	.43	.35	.29
Garamond Light Condensed (ITC) T/D	5.54	4.75	4.16	3.70	3.33	3.02	2.77	2.38	1.85	1.39	1.11	.92	.69	.55	.46
Garamond Light Condensed Italic (ITC) T/D	5.39	4.62	4.04	3.59	3.23	2.94	2.70	2.31	1.80	1.35	1.08	.90	.67	.54	.45
Garamond Book Condensed (ITC) T/D	5.21	4.30	3.91	3.47	3.13	2.85	2.60	2.23	1.74	1.30	1.04	.87	.65	.52	.43
Garamond Book Condensed Italic (ITC) T/D	5.24	4.49	3.93	3.50	3.15	2.86	2.62	2.25	1.75	1.31	1.05	.87	.66	.52	.44
Garamond Bold Condensed (ITC) T/D	4.64	3.97	3.48	3.09	2.78	2.53	2.32	1.99	1.55	1.16	.93	.77	.58	.46	.39
Garamond Bold Condensed Italic (ITC) T/D	4.68	4.01	3.51	3.12	2.81	2.55	2.34	2.00	1.56	1.17	.94	.78	.58	.47	.39
Garamond Ultra Condensed (ITC) T/D	4.62	3.96	3.47	3.08	2.77	2.52	2.31	1.98	1.54	1.16	.92	.77	.58	.46	.38
Garamond Ultra Condensed Italic (ITC) T/D	4.42	3.79	3.31	2.94	2.65	2.41	2.21	1.89	1.47	1.10	.88	.74	.55	.44	.37
Garth Graphic T/D	4.27	3.66	3.20	2.85	2.56	2.33	2.13	1.83	1.42	1.07	.85	.71	.53	.43	.36
Garth Graphic Italic T/D	4.68	4.01	3.51	3.12	2.81	2.55	2.34	2.00	1.56	1.17	.94	.78	.58	.47	.39
Garth Graphic Bold T/D	3.84	3.29	2.88	2.56	2.30	2.09	1.92	1.64	1.28	.96	.77	.64	.48	.38	.32
Garth Graphic Bold Italic T/D	4.16	3.57	3.12	2.77	2.50	2.27	2.08	1.78	1.39	1.04	.83	.69	.52	.42	.35
Garth Graphic Extrabold T/D	3.68	3.15	2.76	2.46	2.21	2.01	1.84	1.58	1.23	.92	.74	.61	.46	.37	.31
Garth Graphic Condensed T/D	5.39	4.62	4.04	3.59	3.24	2.94	2.70	2.31	1.80	1.35	1.08	.90	.67	.54	.45
Garth Graphic Bold Condensed T/D	4.89	4.19	3.67	3.26	2.93	2.67	2.44	2.09	1.63	1.22	.98	.81	.61	.49	.41
Geometric Light T/D	3.59	3.08	2.69	2.39	2.16	1.96	1.80	1.54	1.20	.90	.73	.61	.45	.36	.30
Geometric Light Italic T/D	3.59	3.08	2.69	2.39	2.16	1.96	1.80	1.54	1.20	.90	.73	.61	.45	.36	.30
Geometric Demi T/D	3.59	3.08	2.69	2.39	2.16	1.96	1.80	1.54	1.20	.90	.73	.61	.45	.36	.30
Geometric Demi Italic T/D	3.59	3.08	2.69	2.39	2.16	1.96	1.80	1.54	1.20	.90	.73	.61	.45	.36	.30
Gill Sans Light T/D	4.96	4.25	3.72	3.31	2.98	2.71	2.48	2.13	1.65	1.24	.99	.83	.62	.49	.41
Gill Sans Light Italic T/D	5.28	4.52	3.96	3.52	3.17	2.88	2.64	2.26	1.76	1.32	1.06	.88	.66	.53	.44
Gill Sans Medium T/D	4.91	4.21	3.68	3.27	2.95	2.68	2.46	2.10	1.64	1.23	.98	.82	.61	.49	.41
Gill Sans Medium Italic T/D	5.24	4.49	3.93	3.48	3.14	2.85	2.62	2.24	1.74	1.31	1.04	.87	.65	.52	.44
Gill Sans Bold T/D	4.20	3.60	3.15	2.80	2.52	2.30	2.10	1.80	1.40	1.05	.84	.70	.52	.42	.35
Gill Sans Light Condensed T/D	6.21	5.32	4.65	4.14	3.72	3.39	3.10	2.66	2.07	1.55	1.24	1.03	.78	.62	.52

Appendix B

LINE SPACE CALCULATION

Line Space is the advancement of the photographic paper or film, thereby providing space for the typeset line. The Point Size is the determining factor when selecting Line Space and the depth for the manuscript copy is determined by the total amount of Line Space.

LINE SPACE DEPTH

After the determination of the number of lines that will be attained in a specified style and point size, it is often necessary to space out the lines to fill a given area. Multiplying the number of lines by the type size will determine the actual depth of the type. This does not include any additional space. Subtracting the type size from the required depth in points and dividing by one less than the number of lines will determine the required Line Space. The total will be the Line Space value multiplied by the number of lines, plus the body size of the last line.

PROCEDURES:

Filling Line Space Depth To A Given Area.

1. The number of typeset lines for the manuscript copy is required as well as the depth in picas for the area.
2. Convert the pica depth to points by multiplying by 12. This will determine the given area in points.
3. Subtract the type size, in points, from the total area depth.
4. Divide the total depth in points by one less than the number of lines.
5. Prove the determined Line Space by multiplying the value by the number of lines, plus the point size for the last line.

It has been previously determined that there are 14 lines of 9 Point Garth for the manuscript copy on an 18 pica measure. The copy must now be typeset to occupy a total depth of 14 picas.

14 Lines
× 12
168 Total Depth

168
− 9 Minus Point Size (9)
159

12 Required Line Space
13 ⌊159

Completed Copy:

Drawing a line through the average width of the typewriter copy will determine the average number of characters in a line. Counting the characters in the average line will determine the approximate number of characters in the manuscript copy. Short lines can counted as full lines, since the typeset copy will also have short lines. If there are several pages of text, it may be necessary to fine the average number of lines per page. This is accomplished by adding the total number of lines and dividing by the number of pages. If type style and size are determined in advance, the manuscript copy can be typed with the proper number of characters to the line so that the typesetting will run approximately line-for-line.

Quick Reference To File Management

ABORT

Procedure: *Abort*

APPEND

APPEND PAGE TO END OF A NEW PRIMARY FILE

Procedure: *Limit/Scroll Up* to position cursor at the bottom of the data. *Append/Execute* will enter the page at the end of the file and the system will be addressed to the next sequential page.

APPEND PAGE TO END OF EXISTING PRIMARY FILE

Procedure: *Limit/Scroll Up* to position cursor at the bottom of data. *Append/Execute* will enter the page at the end of the file and the system is addressed to the *current* page.

APPEND CURRENT PAGE TO END OF SECONDARY FILE

Procedure: Move cursor to end of data/*Secondary/Append*/Secondary file name/*Execute*

APPEND PAGE FROM SECONDARY FILE TO PRIMARY

Procedure: *Secondary/Call Page*/File Name/*Execute*/Move cursor to end of data/*Append/Execute*

APPEND/BLOCK

APPEND BLOCK, PRIMARY FILE

Procedure: *Define/Block*/Move cursor to define block/*Execute*
Append/Block/File Name/*Execute*

APPEND BLOCK, SECONDARY FILE

Procedure: *Define/Block*/Move cursor to define block/*Execute*
Secondary/Append/Block/File Name/*Execute*

***Where applicable, the Disk Drive must be identified (F1, F2, etc.)**

CALL PAGE

CALL PAGE

Procedure: *Call Page/Enter file name/Execute*

CALL FILE AND PAGE NUMBER

Procedure: *Call Page*/Enter file name/
Cursor Right/Change page number/*Execute*

CALL PAGE, SECONDARY FILE

Procedure: *Secondary/Call Page/File Name*/Enter page number/*Execute*

CLEAR DISPLAY

Procedure: *Delete/Display/Execute*

CLEAR INDEX

Procedure: *Abort*

CLOSE FILE

Procedure: *Close File/Execute*

DEFINE BLOCK

Procedure: Move cursor to start of block/*Define/Block*/Move cursor to define data/*Execute*

DELETE BLOCK

Procedure: *Define block*/Move cursor to define block/*Execute/Delete/Block/Execute*

DELETE PAGE

Procedure: *Delete*/Page Number/*Execute* (File must be open)

Quick Reference To File Management

DISPLAY COMPOSITION PARAMETERS

Procedure: *File Func/Compose Key/Execute/Scroll Up/Scroll Down (Abort to clear)*

DISPLAY FONTS

Procedure: *Display/*FT/*Execute (Abort to clear)*

DISPLAY INDEX

Procedure: *Display/*DI/Drive No./*Execute (Abort to clear)*

Procedure: *Display/*PR/*Execute (Display Printer Queue)*

Procedure: *Display/*TY/*Execute (Display Typesetter Queue)*

Procedure: *Display/*PV/*Execute (Display Preview Queue)*

Procedure: *Display/*RE/*Execute (Display ACI Receive Queue)*

INSERT PAGE

INSERT PAGE TO PRIMARY FILE

Procedure: *Move cursor to end of current page/Insert/Page No./Execute*

INSERT PAGE TO SECONDARY FILE

Procedure: *Move cursor to end of current page/Secondary/Insert/File Name/Page No./Execute*

MEMORY FULL/DIVIDE PAGE

Procedure: *Move cursor to logical breaking point/Insert//Execute/Replace/Execute*

INSERT BLOCK

Procedure: *Define Block/Insert/Block/*Page No./*Execute*

INSERT BLOCK (Secondary File)

Procedure: *Define Block/Secondary/Insert/Block/*File Name/Page No./*Execute*

***Where applicable, the Disk Drive must be identified (F1, F2, etc.)**

NEW FILE

Procedure: *New File/*Enter new file name/*Execute*

MERGE

MERGE DESIGNATED PAGE TO BOTTOM OF CURRENT PAGE OF PRIMARY FILE

Procedure: *Move cursor to bottom of Current Page/Merge/*Page No./*Execute*

MERGE DESIGNATED PAGE TO TOP OF CURRENT PAGE

Procedure: *Move cursor to top of Current Page/Merge/Page No./Execute*

MERGE PAGE FROM SECONDARY FILE TO BOTTOM OF CURRENT PAGE

Procedure: *Move cursor to bottom of current page/Secondary/Merge/*File Name/Page No./*Execute*

REPLACE PAGE

REPLACE PAGE

Procedure: Call file and curent page for editing. Enter corrections/*Replace Page/Execute*

REPLACE PAGE/CALL PAGE

Procedure: Call file and current page for editing. Enter corrections/*Replace Page/Call Pagexx/Execute*

SEARCH

Procedure: *Search/*Enter Data /*Execute*

SEARCH AND REPLACE

Procedure: *Search/*Enter Data /*Replace/*Enter Data/*Execute*

Quick Reference To System Commands

CALL PRINTER MENU

Procedure: *System Command*/PMEN/*Execute* (See Printer Reference Manual)

CHANGE DISPLAY MAPPING

Procedure: *System Command*/DMAP/Mapping No./*Execute*

CLEAR FONT MEMORY

Procedure: *File Function*/EDIT/*Execute*/*System Command*/CFMEM/*Execute*

COPY FILE*

Procedure: *System Command*/COPF/Enter Source File Name/Move cursor to destination field/Enter New File Name/*Execute*

DELETE FILE*

Procedure: *System Command*/*Delete*/*Disk Drive*/Enter File Name/Extension/*Execute*

DELETE FONT

Procedure: *System Command*/DFNT/Font Identity No./*Execute*

DELETE JOB*

Procedure: *System Command*/DJOB/TY/Job Number/*Execute* (Typesetter)

Procedure: *System Command*/DJOB/PR/Job No./*Execute* (Printer)

Procedure: *System Command*/DJOB/PV/Job No./*Execute* (Preview)

Procedure: *System Command*/DJOB/RE/Job No./*Execute* (ACI Receive Queue)

ERASE DISK

Procedure: *System Command*/EDSK/Enter Disk Volume Name/*Execute*

LOAD FONT

Procedure: *System Command*/LFNT/Enter Font Identity Number/*Execute*

LOAD FONT DRESS*

Procedure: *System Command*/DRESS/Enter Font Dress Name/*Execute*

LOAD UDK*

Procedure: *System Command*/LUDK/File Name/Execute

MERGE UDK*

Procedure: *System Command*/MUDK/File Name/*Execute*

RESET QUEUE/PERIPHERAL DEVICE

Procedure: *System Command*/RES/TY/*Execute* (Reset Typesetter)

Procedure: *System Command*/RES/PR/*Execute* (Reset Printer)

Procedure: *System Command*/RES/PV/*Execute* (Reset Preview)

Procedure: *System Command*/RES/RE/*Execute* (Reset ACI Receive Queue)

SELECT ACI TERMINAL MODE

Procedure: *System Command*/TERM/*Execute* (See ACI Reference Manual)

***Where applicable, the Disk Drive must be identified (F1, F2, etc.)**

Advance Leading . . . . . **l.c. ALD key xxxx.x** (points, ¼-points)
Auto Center Mode . . . . . **AQ key, 'c'**
Auto Left Mode . . . . . **AQ key, 'l'** (l.c. 'ell')
Auto Line End . . . . . **UDK 'aa'**
Auto Right Mode . . . . . **AQ key, 'r'**
AVCC Select . . . . . **UDK 'C' x** (style 1, 2 or 3)
Back Point . . . . . **UDK 'P' xx** (no. of points)
Back Unit . . . . . **UDK 'U' xx** (no. of units)
Call Tab . . . . . **Call Tab ##, or Tab key**
Change Letterspace . . . . . **UDK 'L' xx.xx** (positive/negative)
Change Set Size . . . . . **UDK 'w' xx.x** (points, ½-points)
Change Word Space . . . . . **UDK 's' xx.xx.xx** (min./pref./max.)
Char. Comp. ON . . . . . **l.c. CCO key**
Char. Comp. OFF . . . . . **u.c. CCX key**
Char. Comp. Set . . . . . **UDK 'c' xx** (9th points)
Cut Left . . . . . **UDK 'o'#,000.00,000.00,00.00**
Cut Right . . . . . **UDK 'O'#,000.00,000.00,00.00**
Deactivate Tabs . . . . . **CR key** (or PREC/RETURN)
Define Pseudo Italic . . . . . **UDK 'v' xx** (degree)
Define Tab . . . . . **UDK 't'##,xx.xx,xx.xx,Q**
Define Floating Tab . . . . . **UDK 'T'F##,xx.xx,Q**
Flash Only . . . . . **UDK 'f' or 'F'**
Forward Point . . . . . **UDK 'p' xx** (no. of points)
Forward Unit . . . . . **UDK 'u' xx** (no. of units)
Font Select . . . . . **FT key, xxxx** (1 to 4 digits)
Hyphenation OFF . . . . . **UDK 'H'**
Hyphenation ON . . . . . **UDK 'h'**

Inhibit Hyphenation . . . . . **DH before or after word**
Insert Character . . . . . **INS CHAR key + character**
Insert Rule . . . . . **INS BLR key**
Insert Space . . . . . **INS SPACE key**
Kerning On . . . . . **UDK 'k'**
Kerning Off . . . . . **UDK 'K'**
Letterspace ON/OFF . . . . . **UDK 'l' x** (o = ON; x = OFF)
Line Length . . . . . **LL key xx.xx** (picas/points)
Line Space . . . . . **LS key xxx.x** (points/¼-points)
Manual Line End . . . . . **UDK 'a' 'm'**
Minus Line Space . . . . . **u.c. MLS key**
Minus 1-Point Lead . . . . . **u.c. MPL key**
No Flash ON . . . . . **UDK 'n'**
No Flash OFF . . . . . **UDK 'N'** (or line-end command)
Plus Line Space . . . . . **l.c. PLS key**
Plus 1-Point Lead . . . . . **l.c. PPL key**
Point Size . . . . . **PS key xx.x** (points/½-points)
Pseudo Italic Define . . . . . **UDK 'V' xx** (degree of slant)
Pseudo Italic ON/OFF . . . . **UDK 'v' x** (o = ON; x = OFF)
Reverse Lead . . . . . **u.c. RLD key xxxx.x** (points/¼-points)
Semi-Auto Line-End . . . . . **UDK 'a' 's'**
Set Both-Side Indent . . . . . **UDK 'I' 'b' xx.xx** (pica/point indent)
Set Hanging Indent . . . . . **UDK 'I' 'h' xx.xx** (pica/point indent)
Set Left Indent . . . . . **UDK 'I' 'l' xx.xx** (pica/point indent)
Set Right Indent . . . . . **UDK 'I' 'r' xx.xx** (pica/point indent)
Turn Indent ON/OFF . . . . . **UDK 'i'** (o = ON; x = OFF)

INTRODUCTION

The Vertical Justification Option allows the user to automatically vertically justify columns of text. The user designates where, within the column, the excess vertical white space will be placed using the following commands.

COMMANDS:

<VJOddd.dd,q>

ddd.dd Designates the depth (picas and points) of the vertical column after the vertical justification is complete. The base line at the point of insertion defines the top of the column.

q **Vertical Justification Mode:** **U** (UP), **D** (Down), **C** Center, **J** (Justify-also the default).

<VJX>

Defines the end of the column of text to be vertically justified. The line in which the VJO command is placed will be used to calculate the deepest vertical position. The base line the point of insertion determines the bottom of the VJ column.

<VISn>

Defines the point(s) where excess vertical space will be inserted during vertical justification.

n (Optional) A proportional value used for more control over excess space distribution. For example, a <VIS2> command will expand twice as much as a <VIS> or a <VIS1> (default).

DISPLAY PARAMETERS

The position of excess vertical white space can be examined at the point of the cursor by using the DISPLAY PARAMETERS function of the MCS. The following parameters are included which pertain to the VJ option:

V Quad Mode:

The vertical quad mode presently in effect (J, U, D, C or X if VJ is not in use).

VIS Count: aaaa,nnnn

The total number of VIS commands (a), and the total number of proportional values (n) presently in use.

Excess V Space:

The amount of vertical space remaining in the Vertical Justification Column.

Vertical Justification Option

EXAMPLES

<VJO10,J>

TROPICAL & MARINE FISH • AFRICANS • BIRDS
SMALL ANIMALS • RARE & COMMON REPTILES
AKC PUPS • KITTENS

<VIS>

MANSFIELD
AQUARIUM - PET SHOP
COMPLETE LINE OF PET SUPPLIES

<VIS>

253 SOUTH ST.
MANSFIELD, MA

356-1234
We Can Obtain The Unusual

<VJX>

The business card on the left is an example of vertical justification in the justified mode.

<VJO8,C>

This example demonstrates the use of the Vertical Justification *centering* mode. This column is vertically centered between the two horizontal rules using the VJ centering command as illustrated.

<VJX>

<VJO08,U>

This example demonstrates the use of the Vertical Justification *quad up* mode. This column is vertically quadded up between the two horizontal rules using the VJ quad up command as illustrated.

<VJX>

<VJO08,D>

This example demonstrates the use of the Vertical Justification *quad down* mode. This column is vertically quadded down between the two horizontal rules using the VJ quad down command as illustrated.

<VJX>

RESTRICTIONS:

A vertically justified column can be defined almost anywhere in the text stream however, there are some restrictions that, if encountered, will cause operator errors and will be reported.

An MCO command may not occur between Vertical Justify ON/OFF commands.

Only one VJO command may be active at any one time.

Plus Point Lead, Plus Line Space, Advance Lead may be used in a VJ column, however if execution of these commands cause advance leading deeper than the bottom of the VJ column, an error will be reported.

Minus Point Lead, Minus Line Space and Reverse Lead may be used in a VJ column, however if execution of these commands cause reverse leading above the start of the VJ column, an error will be reported.

Cut run arounds may not to be used in the vertical justification column.

When setting blocks of negative video copy within a VJ column, care must be taken to avoid white spaces between the lines in the blocks after vertical justification.

Index

A

Abort, 16
Advance Leading, 70
Alternate Hyphenation, 62
Append, 31, 42-43, 101, 111
Appendix A, 101
Appendix B, 103-110
Auto Quadding, 63
Automatic Tabs, 73-74
AVCC, 69, 99

B

Back Point, 83
Back Unit, 83
Blocks, 50,

C

Call Page, 45, 103, 111
Cancel Character, 24
Cancel Word, 24
Character Compensation, 69
Character Complements, 95
Character Count, 107
Cicero Conversion, 101
Clear Display, 15, 111
Close File, 44
Composition Mode, 25-26
Control Keypads, 4-5
Copy File, 102
Copy Fitting, 107-110
Copy Preparation, 103
Cursor, 16-23
Cut Run-Arounds, 79

D

Data Entry, 25-27, 90
Define Block, 30
Delete Block, 31
Delete Character, 28
Delete File, 106
Delete Line, 28
Delete Page, 29
Discretionary Hyphen, 62
Disk Drives, 3, 39
Display Index, 41, 102
Display Mapping, 95
Display Parameters, 54
Display UDK, 94

E

Editing Keypad, 4, 16-36
Edit Mode, 27
End Line, 61
Erase Disk, 106
Execute, 40

F

File Management, 5, 37-52,
Fixed Spaces, 67
Flash Only, 81
Floating Tabs, 71
Font Loading, 12
Font Parameters, 56
Forward Point, 82
Forward Unit, 82

H

Hyphenation, 62

I

Indents, 77-78
Index, 117
Insert Baseline Rule, 70
Insert Block, 50
Insert Character, 76
Insert Mode, 28
Insert Page, 48-49, 105
Insert Space, 75
Interword Space, 65
Introduction, 1-12

J

Justification, 62

K

Kerning, 98
Keyboard Arrangements, 8-11

L

Letterspacing, 66
Line Ending Modes, 60-61
Line Length, 55
Line Space,
53, 59, 70, 104-105, 110
Load Button, 3
Loading Fonts, 12
Local Compose Mode, 27

M

Manual Quadding, 64
Markup, 106-109
Memory Full, 48
Merge, 51-52, 104
Merge UDK File, 94
Metric Conversion, 101
Mini Floppy Disk, 2, 37
Minus Line Space, 70
Multi Disk Option, 100-106

N

New File, 40
Next Page, 46, 51
No Flash, 80

P

Parameter Display, 54
Pica Conversion, 101
Plus Line Space, 70
Point Size, 53, 57
Precedent Characters, 7, 68
Previous Page, 46, 52
Primary Mode, 38
Protected Files, 45
Pseudo Italic, 84

Q

Quadding, 63-64
Queue, Typesetter, 96-97
Quick References To:
File Management, 111-117
System Commands, 113
Typeset Commands, 114

R

Replace Page, 47
Reverse Line Space, 70
Reverse Video, 85
Return, 61

S

Scrolling, 21-23
Search and Replace, 32-36
Secondary Mode, 38
Set Size, 58
Set Spaceband, 65
Status Lines, 13-14
System Commands, 113
System Controls, 3

T

Tabbing, 71-74
Terminal Program, 12
Time Out, 15
Typesetter Commands, 53-99
Typesetter Queue, 96-97
Typewriter Keyboard, 6-11

U

User Define Keys, 76, 86-94

V

Vertical Justification, 115-116
Video Terminal, 1, 12-23
Videoscreen, 13-15

W

Word Wrap, 15

200 Ballardvale Street
Wilmington, MA 01887